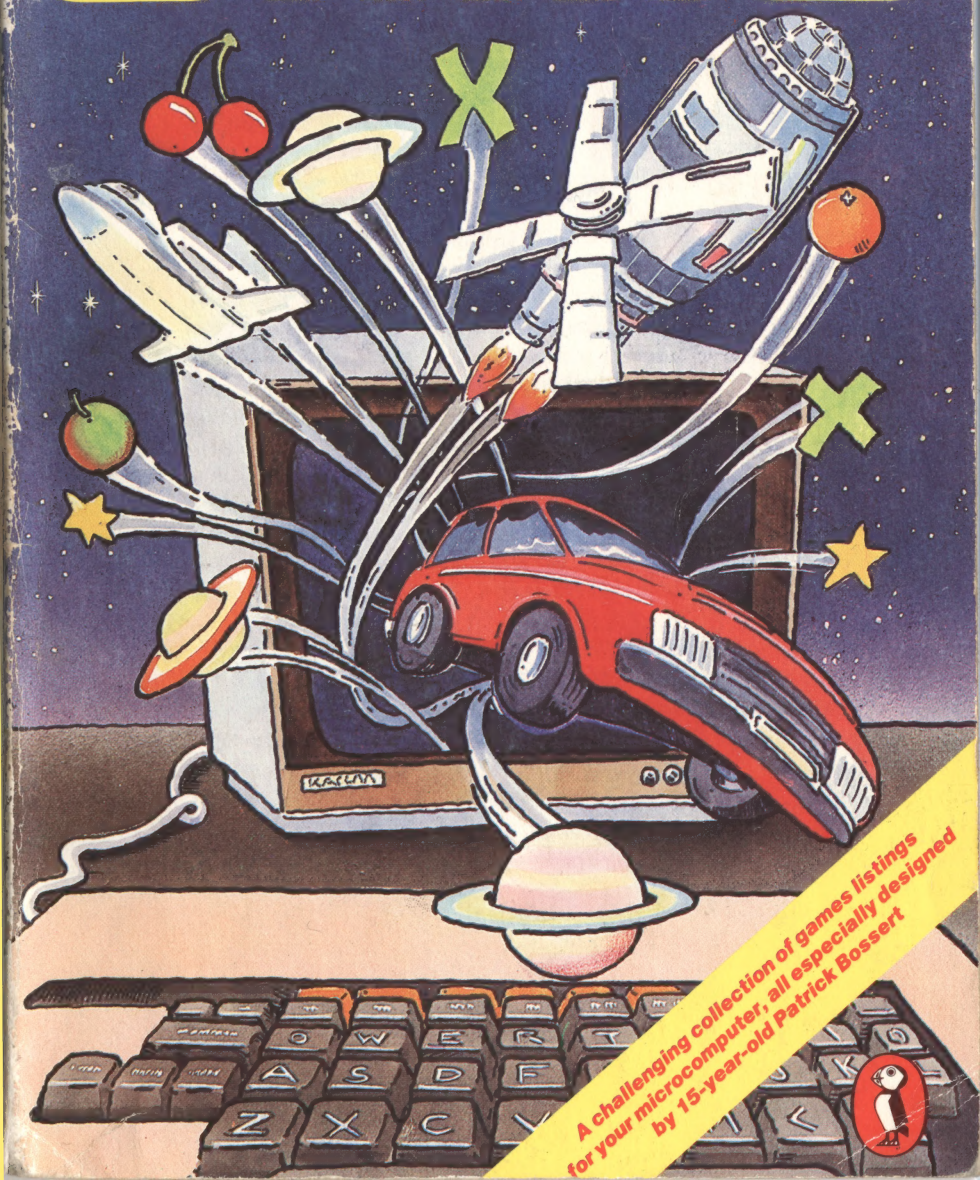
PATRICK BOSSERT AND PHILIPPA DICKINSON

MICRO GAMES



A challenging collection of games listings for your microcomputer, all especially designed by 15-year-old Patrick Bossert

Puffin Books

Micro Games

Computer games are great to play, and the pleasure is even greater when *you* programmed the game in the first place! Not only will you know exactly how the game works but you can also make up all kinds of alterations and adjustments so that it stays fresh and challenging for a long time.

Patrick Bossert and Philippa Dickinson (both of whom have previously published books on games and puzzles) have devised thirteen fast and exciting computer games which you can program into your own personal or school micro. The programs are explained step by step, and general advice is given on how to make each game even more difficult once you have completed the program successfully.

So when you have worked your way through the book, you will have a batch of great games to play (they can be stored easily on cassette), and you will also have learnt exactly how to handle your computer. The book can be used with the following microcomputers: BBC (models A and B), RML (380Z and 480Z), ZX Spectrum and ZX81 (with 16K RAM).

Patrick Bossert is the author of the phenomenally successful *You Can Do the Cube*, which in 1981 showed millions of frustrated people around the world – adults and children alike – how to solve Rubik's Cube. Patrick is now fifteen and more interested in computers than cubes. **Philippa Dickinson** is the author of *Go! A Book of Games* (also published by Puffin).

Patrick Bossert and Philippa Dickinson

Micro Games

Illustrated by John Farman

Puffin Books

Puffin Books, Penguin Books Ltd, Harmondsworth, Middlesex, England
Penguin Books, 40 West 23rd Street, New York, New York 10010, U.S.A.
Penguin Books Australia Ltd, Ringwood, Victoria, Australia
Penguin Books Canada Ltd, 2801 John Street, Markham, Ontario, Canada L3R 1B4
Penguin Books (N.Z.) Ltd, 182–190 Wairau Road, Auckland 10, New Zealand

First published 1983

Copyright © Patrick Bossert and Philippa Dickinson, 1983
Illustrations copyright © John Farman, 1983
All rights reserved

Printed in Great Britain by
Richard Clay (The Chaucer Press) Ltd,
Bungay, Suffolk
Set in Linotron Helvetica and Palatino by
Rowland Phototypesetting Ltd, Bury St Edmunds, Suffolk

Except in the United States of America, this book is sold subject
to the condition that it shall not, by way of trade or otherwise, be lent,
re-sold, hired out, or otherwise circulated without the
publisher's prior consent in any form of binding or cover other than
that in which it is published and without a similar condition
including this condition being imposed on the subsequent purchaser

Contents

Introduction 7

Bouncers 12
BBC, ZX Spectrum, RML, ZX81

Breakout 16
BBC, ZX Spectrum, RML

Mastermind 28
ZX81

Asteroid Belt 32
BBC, ZX Spectrum, RML

Danger Orbit 39
BBC, ZX Spectrum, RML, ZX81

One-way Street 52
BBC, ZX Spectrum, RML

Memory Mix 59
ZX81

Miner 63
BBC, ZX Spectrum, RML

Shooting in the Dark 74
ZX81

Snake Runner 78
BBC, ZX Spectrum, RML

Magic Square 89
BBC, ZX Spectrum, RML, ZX81

Pickup 101
BBC, ZX Spectrum, RML

Fruit Machine 113
BBC, ZX Spectrum, RML, ZX81

Writing Instructions 131

Writing Your Own Programs 133

Notes 135

Introduction

The chief attraction of micro games like these is that you program them yourself. Unlike video games, or pre-packaged software, you will know exactly how the games work and can make any amount of alterations to them. Writing games programs is also one of the most enjoyable ways of learning the capabilities of your micro and getting a real feeling for BASIC.

There are games here for the BBC (models A and B), RML (380Z and 480Z), ZX Spectrum and ZX81 (with 16K RAM). The ZX81 is much slower than the other micros and cannot manage a few of the very fast games in this book, so we have devised three games especially for it.

Each of the games in this book has been broken down into small chunks and explained in detail, which will make copying easy and also help you pick out different elements to use when writing your own programs.

Be very careful when you are copying a listing, as the slightest mistake, just a comma missed out, could prevent the program from running. Use a ruler or a sheet of paper to mark your place and check each line as you go. Remember to press RETURN or ENTER at the end of each program line. ZX Spectrum and ZX81 users should remember to use the command keys where appropriate. If the program includes an ON ERROR GOTO . . . statement, leave this out until you are sure that the game runs correctly.

If the program won't work, don't despair. You have almost certainly made an error in the copying. Get your computer to list the program for you and check it line by line for mistakes.

Here is a list of common mistakes to look out for:

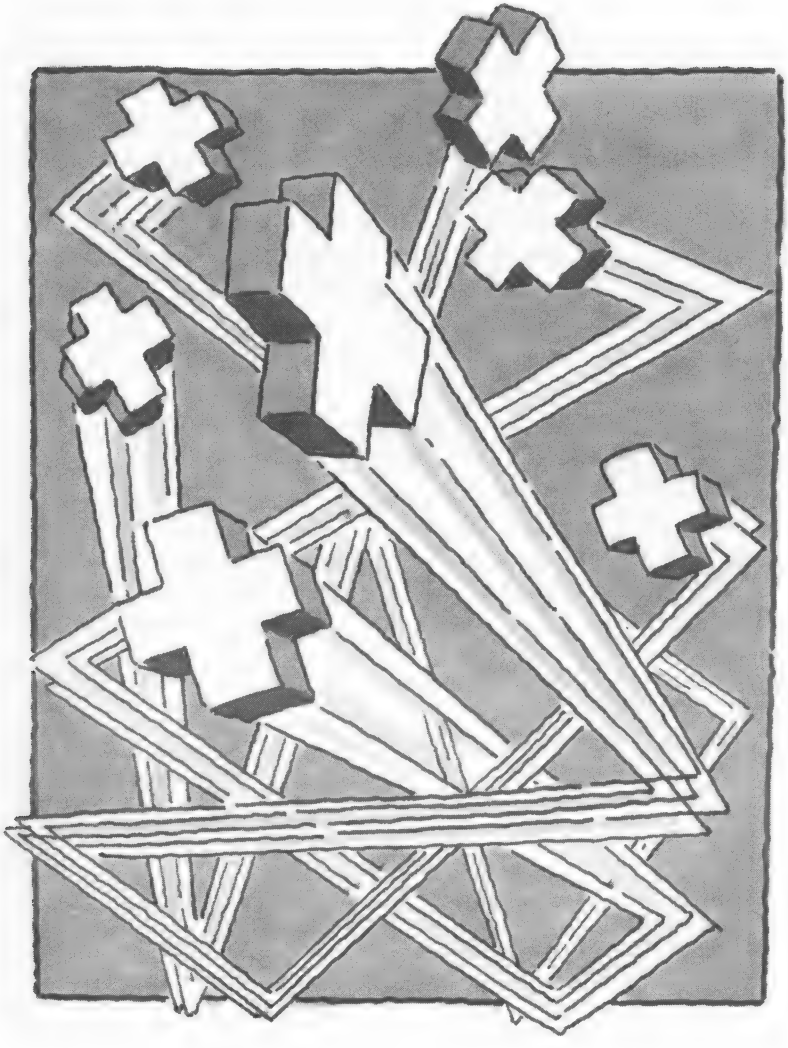
1. Line missed out.
2. Two lines joined together (you forgot to press RETURN or ENTER after the first line).
3. Commas, colons, semi-colons or brackets missing or in the wrong place.
4. Wrong symbols used – it is easy to mistake Ø for O when copying, also easy to press the SHIFT key by mistake when typing a symbol.
5. Spaces missing or in the wrong place – e.g., no space before bracket in DIM A(7).
6. Line copied from wrong program.

Once you have made the program run successfully, you could save it on cassette (your manual will give instructions for this) and then start playing around with the listing. In some cases we have given suggestions for ways to improve or change the programs, which you can try. Don't be afraid to experiment – you can't do your computer any harm and you can always return to the original listing if you want to. If your computer has colour and sound, make as much use of these facilities as you can. Add them to the existing programs and use them when writing your own.

Micro Games

Bouncers

BBC, ZX Spectrum, RML, ZX81



Two little programs to get you started. So many computer games rely on a ball bouncing back and forth from bat to wall or something similar that we decided to show you how it works. The first of these programs makes a + bounce from left to right and back, over and over again. The second program goes one further, sending the + bouncing all over the screen.

The Programs

The first of these programs is very simple. X and Y are the coordinates of the +, while XD is its direction. Line 50 prints the +, line 60 gives a very short delay before line 70 unprints the +. The computer checks whether the bouncer has hit either side of the screen in line 80 and, if so, changes its direction. Line 90 gives the bouncer a new x-coordinate, and line 100 loops back to line 50 to print it again.

BBC

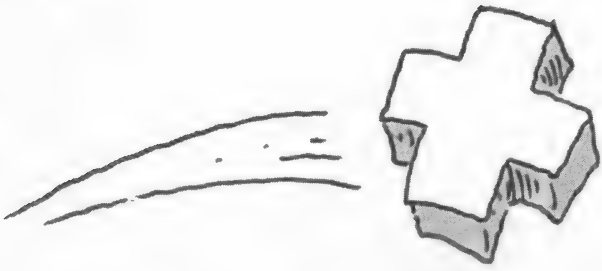
```
 10 MODE7
 20 VDU23;8202;0;0;0;
 30 LET X=1:LET XD=1
 40 LET Y=10
 50 PRINTTAB(X,Y);"+"
 60 FORR=1 TO 29:NEXT R
 70 PRINTTAB(X,Y);" "
 80 IF X<1 OR X>30 THEN LET XD=-XD
 90 LET X=X+XD
100 GOTO 50
```

Spectrum and ZX81

```
 10 REM
 20 REM
 30 LET X=1:LET XD=1
 40 LET Y=10
 50 PRINT AT Y,X;"+"
 60 FORR=1 TO 5:NEXT R
 70 PRINT AT Y,X;" "
 80 IF X<1 OR X>30 THEN LET XD=-XD
 90 LET X=X+XD
100 GOTO 50
```

RML

```
 10 PUT 12
 20 REM
 30 LET X=2:LET XD=2
 40 LET Y=20
 50 PLOT X,Y,"+"
 60 FORR=1 TO 29:NEXT R
 70 PLOT X,Y," "
 80 IF X<2 OR X>60 THEN LET XD=-XD
 90 LET X=X+XD
100 GOTO 50
```



The second of these two programs brings in the added element of a changing y-coordinate, making the + bounce all over the screen. The principle is just the same, only we now have to consider YD, the up-and-down direction of the bouncer. Line 90 checks whether the bouncer has hit either the top or the bottom of the screen and, if so, changes its direction and gives it a new y-coordinate (line 110).

BBC

```
 10 MODE7
 20 VDU23;8202;0;0;0;
 30 LET X=1:LET XD=1
 40 LET Y=10:LET YD=1
 50 PRINTTAB(X,Y);"+"
 60 FORR=1 TO 29:NEXT R
 70 PRINTTAB(X,Y);" "
 80 IF X<1 OR X>30 THEN LET XD=-XD
 90 IF Y<1 OR Y>20 THEN LET YD=-YD
100 LET X=X+XD
110 LET Y=Y+YD
120 GOTO 50
```

Spectrum and ZX81

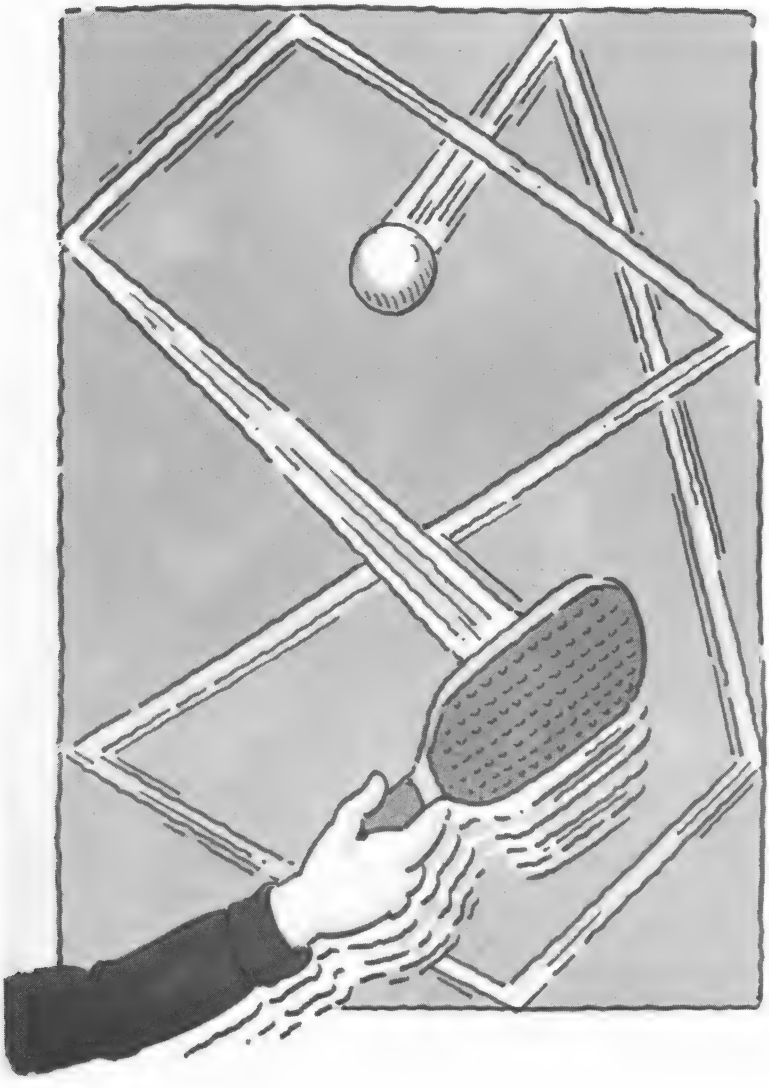
```
 10 REM
 20 REM
 30 LET X=1:LET XD=1
 40 LET Y=10:LET YD=1
 50 PRINT AT Y,X;"+"
 60 FORR=1 TO 5:NEXT R
 70 PRINT AT Y,X;" "
 80 IF X<1 OR X>30 THEN LET XD=-XD
 90 IF Y<1 OR Y>20 THEN LET YD=-YD
100 LET X=X+XD
110 LET Y=Y+YD
120 GOTO 50
```

RML

```
 10 PUT 12
 20 REM
 30 LET X=2:LET XD=2
 40 LET Y=20:LET YD=3
 50 PLOT X,Y,"+"
 60 FORR=1 TO 29:NEXT R
 70 PLOT X,Y," "
 80 IF X<2 OR X>60 THEN LET XD=-XD
 90 IF Y<3 OR Y>57 THEN LET YD=-YD
100 LET X=X+XD
110 LET Y=Y+YD
120 GOTO 50
```

Breakout

BBC, ZX Spectrum, RML



A great demolition game! You have a bat and a ball and are aiming to break down the wall, four thicknesses deep, in front of you. Each time your ball hits the wall, it knocks out a brick. Sometimes it will knock out even more. Quick reactions are needed to make sure that your bat fields the ball as it bounces back from the wall – it is all too easy to miss! You get five balls to play with and the game gets harder and harder as the holes get larger and larger.

Player keys

z – left
x – right

The Program

Let's begin by giving the computer some initial data to work from (line 10). BX is the x-coordinate of the bat, A$ is the bat itself (the Spectrum uses graphic shift 3 for four character spaces), YD is the direction of the ball, SC is your score and BALL tells you how many balls you have used. Line 20 gives dimensions to the array B, which defines the area of the game and gives each position within that array the value of zero.

BBC

```
   10 LET BX=14:LET A$="££££":LET YD=1:LET SC=0:LET BALL
=1
   20 DIMB(37,24)
   30 *FX11,1
   40 *FX12,1
   50 MODE7
   60 VDU23;8202;0;0;0;
```

Spectrum

```
   10 LET BX=14:LET A$="▬▬▬":LET YD=1:LET SC=0:LET BALL
=1
   20 DIMB(37,24)
   30 REM
   40 REM
   50 BORDER 0:PAPER 0:CLS
   60 REM
```

RML

```
   10 LET BX=14:LET A$=CHR$(195)+CHR$(195)+CHR$(195)+CHR
$(195):LET YD=1:LET SC=0:LET BALL=1
   20 DIMB(37,24)
   30 REM
   40 REM
   50 PUT 12
   60 REM
```

Then we print the game court. Line 70 gives each character in the four walls a value of 1 (a device that is used to make knocking out a brick fairly simple). On the BBC we can make the ball change colour in line 80. The walls are printed at line 90 and the bat at line 100. Line 120 prints the ball (the Spectrum uses graphic shift 8 for this). Finally, line 140 prints up which ball you are on.

BBC

```
   70 FORR=4 TO 7:FORC=1 TO 33:LET B(C,R)=1:NEXT C:NEXT
R
   80 FORR=1 TO 25:F=R MOD 7:PRINTTAB(0,R);CHR$(145+F):N
EXT R
   90 FORR=4 TO 7:FORC=1 TO 31:PRINTTAB(C,R);CHR$255:NEX
T C:NEXT R
  100 PRINTTAB(BX,19);A$
  110 LET Y=10:LET X=BX+2:LET XD=RND(1)/2+0.5:LET PAUSE=
30
  120 PRINTTAB(X,Y);"/"
  130 FORR=1 TO 1200:NEXT R
  140 PRINTTAB(0,0);"BALL ";BALL
```

Spectrum

```
   70 FORR=4 TO 7:FORC=1 TO 33:LET B(C+1,R)=1:NEXT C:NEX
T R
   80 REM
   90 FORR=4 TO 7:INK R:FORC=1 TO 31:PRINT AT R,C;"■":NE
XT C:NEXT R
  100 INK 3:PRINT AT 19,BX;A$
  110 LET Y=10:LET X=BX+2:LET XD=RND/3+0.2:LET PAUSE=3
  120 INK 3:PRINT AT Y,X;"■"
  130 FORR=1 TO 200:NEXT R
  140 INK 7:PRINT AT 0,0;"BALL ";BALL
```

RML

```
   70 FORR=4 TO 7:FORC=1 TO 33:LET B(C,R)=1:NEXT C:NEXT
R
   80 REM
   90 FORR=4 TO 7:FORC=1 TO 31:PLOT C*2,(21-R)*3,255:NEX
T C:NEXT R
  100 PLOT BX*2,6,A$
  110 LET Y=10:LET X=BX+2:LET XD=RND(1)/2+0.5:LET PAUSE=
30
  120 PLOT X*2,(21-Y)*3,207
  130 FORR=1 TO 1200:NEXT R
  140 PLOT 0,57,BALL+250
```

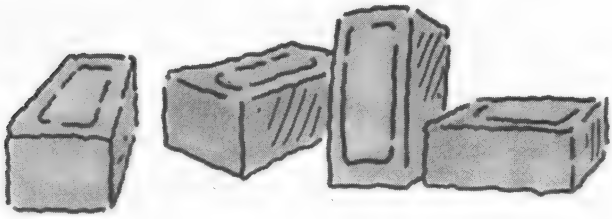
Now we come to the main program loop. Line 150 prints the ball again and line 170 jumps out to a subroutine to move the bat. The ball is unprinted (line 180) and then the computer checks whether the ball has hit a wall by reading the value of B (X, Y). If it is 1 there is a brick there, if 0 there isn't (line 190). If it has hit a brick, the ball changes direction, the brick is knocked out, your score goes up and the computer checks your score in case you have reached the maximum of 128. In lines 210 and 220, the computer checks whether the ball has hit either the top or the sides of the screen and, if so, changes its direction. If the ball has hit the bat (lines 230–240), the computer jumps to lines 250–270 to change the ball's coordinates, before going back to the start of the main loop again.

BBC

```
  150 PRINTTAB(X,Y);"/"
  160 FORR=1 TO PAUSE:NEXT R
  170 GOSUB 290
  180 PRINTTAB(X,Y);" "
  190 IF B(X,Y)=1 THEN LET YD=-YD:SOUND1,-10,125,1:LET B
(X,Y)=0:LET SC=SC+1:PRINTTAB(20,0);"SCORE ";SC*10:IF SC=
128 THEN GOTO 50
  200 IF Y=4 THEN LET PAUSE=1
  210 IF X<2 OR X>31 THEN LET XD=-XD
  220 IF Y<2 THEN LET YD=-YD
  230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
  240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD*0.9:LET YD=-YD
  250 IF Y>20 THEN GOTO 390
  260 LET X=X+XD
  270 LET Y=Y+YD
  280 GOTO 150
```

Spectrum

```
  150 INK 3:PRINT AT Y,X;"■"
  160 FORR=1 TO PAUSE:NEXT R
  170 GOSUB 290
  180 PRINT AT Y,X;" "
  190 IF B(X+1,Y)=1 THEN LET YD=-YD:BEEP 0.1,30:LET B(X+
1,Y)=0:LET SC=SC+1:PRINT AT 0,20;"SCORE ";SC*10:IF SC=12
8 THEN GOTO 50
  200 IF Y=4 THEN LET PAUSE=1
  210 IF X<2 OR X>31 THEN LET XD=-XD
  220 IF Y<2 THEN LET YD=-YD
  230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
  240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD:LET YD=-YD
  250 IF Y>20 THEN GOTO 390
  260 LET X=X+XD
  270 LET Y=Y+YD
  280 GOTO 150
```



RML

```
  150 PLOT X*2,(21-Y)*3,207
  160 FORR=1 TO PAUSE:NEXT R
  170 GOSUB 290
  180 PLOT X*2,(21-Y)*3,32
  190 IF B(X,Y)=1 THEN LET YD=-YD:LET B(X,Y)=0:LET SC=SC
+1:IF SC=124 THEN GOTO 50
  200 IF Y=4 THEN LET PAUSE=1
  210 IF X<2 OR X>31 THEN LET XD=-XD
  220 IF Y<2 THEN LET YD=-YD
  230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
  240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD*0.9:LET YD=-YD
  250 IF Y>=20 THEN GOTO 390
  260 LET X=X+XD
  270 LET Y=Y+YD
  280 GOTO 150
```

We jump to this bat-moving subroutine from line 170 in the main loop. First of all the bat is unprinted (line 310), the computer checks which key you are pressing and changes the bat's x-coordinates accordingly (lines 330–340). The y-coordinate is, of course, constant. Lines 350 and 360 prevent the bat going off the screen, line 370 reprints the bat at its new coordinates and line 380 returns you to the main loop again.

BBC

```
  290 REM move bat routine
  300 *FX15,1
  310 PRINTTAB(BX,19);"     "
  320 LET G$=INKEY$(2)
  330 IF G$="Z" THEN LET BX=BX-2
  340 IF G$="X" THEN LET BX=BX+2
  350 IF BX<1 THEN LET BX=1
  360 IF BX>27 THEN LET BX=27
  370 PRINTTAB(BX,19);A$
  380 RETURN
```

Spectrum

```
290 REM move bat routine
300 REM
310 PRINT AT 19,BX;"    "
320 LET G$=INKEY$
330 IF G$="Z" THEN LET BX=BX-2
340 IF G$="X" THEN LET BX=BX+2
350 IF BX<1 THEN LET BX=1
360 IF BX>27 THEN LET BX=27
370 INK 2:PRINT AT 19,BX;A$
380 RETURN
```

RML

```
290 REM move bat routine
300 REM
310 PLOT BX*2,6,"    "
320 LET G$=GET$(2)
330 IF G$="Z" THEN LET BX=BX-2
340 IF G$="X" THEN LET BX=BX+2
350 IF BX<1 THEN LET BX=1
360 IF BX>27 THEN LET BX=27
370 PLOT BX*2,6,A$
380 RETURN
```

If your ball misses the bat, line 250 sends you to this little routine. Line 400 adds 1 to the number of balls you have used and line 420 checks whether you have used all five before sending you on to the end routine at lines 470–500.

BBC

```
390 *FX15,1
400 LET BALL=BALL+1
410 SOUND1,-15,1,20
420 IF BALL>5 THEN GOTO 440
430 GOTO 110
440 *FX12,4
450 *FX11,66
460 *FX15,1
470 PRINT"NEW GAME ?";
480 INPUT D$
490 IF D$="N" THEN END
500 RUN
```

Spectrum

```
390 REM
400 LET BALL=BALL+1
410 BEEP 1,1
420 IF BALL>5 THEN GOTO 440
430 GOTO 110
440 REM
450 REM
460 REM
470 PRINT"NEW GAME ?";
480 INPUT D$
490 IF D$="N" THEN END
500 RUN
```

RML

```
390 REM
400 LET BALL=BALL+1
410 REM
420 IF BALL>5 THEN GOTO 440
430 GOTO 110
440 REM
450 REM
460 REM
470 PRINT"SCORE=";SC;",NEW GAME ?";
480 INPUT D$
490 IF D$="N" THEN END
500 RUN
```

Suggestions

You could make the game harder by adding another wall. This would not only give you more bricks to aim at but also decrease the distance the ball has to travel, so you would have to react faster. Remember that you would also have to change the total number of bricks in line 190.

Full Program: BBC

```
   10 LET BX=14:LET A$="£££":LET YD=1:LET SC=0:LET BALL
=1
   20 DIMB(37,24)
   30 *FX11,1
   40 *FX12,1
   50 MODE7
   60 VDU23;8202;0;0;0;
   70 FORR=4 TO 7:FORC=1 TO 33:LET B(C,R)=1:NEXT C:NEXT
R
   80 FORR=1 TO 25:F=R MOD 7:PRINTTAB(0,R);CHR$(145+F):N
EXT R
   90 FORR=4 TO 7:FORC=1 TO 31:PRINTTAB(C,R);CHR$255:NEX
T C:NEXT R
  100 PRINTTAB(BX,19);A$
  110 LET Y=10:LET X=BX+2:LET XD=RND(1)/2+0.5:LET PAUSE=
30
  120 PRINTTAB(X,Y);"/"
  130 FORR=1 TO 1200:NEXT R
  140 PRINTTAB(0,0);"BALL ";BALL
  150 PRINTTAB(X,Y);"/"
  160 FORR=1 TO PAUSE:NEXT R
  170 GOSUB 290
  180 PRINTTAB(X,Y);" "
  190 IF B(X,Y)=1 THEN LET YD=-YD:SOUND1,-10,125,1:LET B
(X,Y)=0:LET SC=SC+1:PRINTTAB(20,0);"SCORE ";SC*10:IF SC=
128 THEN GOTO 50
  200 IF Y=4 THEN LET PAUSE=1
  210 IF X<2 OR X>31 THEN LET XD=-XD
  220 IF Y<2 THEN LET YD=-YD
  230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
  240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD*0.9:LET YD=-YD
  250 IF Y>20 THEN GOTO 390
  260 LET X=X+XD
  270 LET Y=Y+YD
  280 GOTO 150
  290 REM move bat routine
  300 *FX15,1
  310 PRINTTAB(BX,19);"     "
  320 LET G$=INKEY$(2)
  330 IF G$="Z" THEN LET BX=BX-2
  340 IF G$="X" THEN LET BX=BX+2
  350 IF BX<1 THEN LET BX=1
  360 IF BX>27 THEN LET BX=27
  370 PRINTTAB(BX,19);A$
  380 RETURN
  390 *FX15,1
  400 LET BALL=BALL+1
  410 SOUND1,-15,1,20
  420 IF BALL>5 THEN GOTO 440
  430 GOTO 110
```

```
  440 *FX12,4
  450 *FX11,66
  460 *FX15,1
  470 PRINT"NEW GAME ?";
  480 INPUT D$
  490 IF D$="N" THEN END
  500 RUN
```

Full Program: Spectrum

```
   10 LET BX=14:LET A$="▬▬":LET YD=1:LET SC=0:LET BALL
=1
   20 DIMB(37,24)
   30 REM
   40 REM
   50 BORDER 0:PAPER 0:CLS
   60 REM
   70 FORR=4 TO 7:FORC=1 TO 33:LET B(C+1,R)=1:NEXT C:NEX
T R
   80 REM
   90 FORR=4 TO 7:INK R:FORC=1 TO 31:PRINT AT R,C;"■":NE
XT C:NEXT R
  100 INK 3:PRINT AT 19,BX;A$
  110 LET Y=10:LET X=BX+2:LET XD=RND/3+0.2:LET PAUSE=3
  120 INK 3:PRINT AT Y,X;"■"
  130 FORR=1 TO 200:NEXT R
  140 INK 7:PRINT AT 0,0;"BALL ";BALL
  150 INK 3:PRINT AT Y,X;"■"
  160 FORR=1 TO PAUSE:NEXT R
  170 GOSUB 290
  180 PRINT AT Y,X;" "
  190 IF B(X+1,Y)=1 THEN LET YD=-YD:BEEP 0.1,30:LET B(X+
1,Y)=0:LET SC=SC+1:PRINT AT 0,20;"SCORE ";SC*10:IF SC=12
8 THEN GOTO 50
  200 IF Y=4 THEN LET PAUSE=1
  210 IF X<2 OR X>31 THEN LET XD=-XD
  220 IF Y<2 THEN LET YD=-YD
  230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
  240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD:LET YD=-YD
  250 IF Y>20 THEN GOTO 390
  260 LET X=X+XD
  270 LET Y=Y+YD
  280 GOTO 150
  290 REM move bat routine
  300 REM
  310 PRINT AT 19,BX;"    "
  320 LET G$=INKEY$
  330 IF G$="Z" THEN LET BX=BX-2
```

```
 340 IF G$="X" THEN LET BX=BX+2
 350 IF BX<1 THEN LET BX=1
 360 IF BX>27 THEN LET BX=27
 370 INK 2:PRINT AT 19,BX;A$
 380 RETURN
 390 REM
 400 LET BALL=BALL+1
 410 BEEP 1,1
 420 IF BALL>5 THEN GOTO 440
 430 GOTO 110
 440 REM
 450 REM
 460 REM
 470 PRINT"NEW GAME ?";
 480 INPUT D$
 490 IF D$="N" THEN END
 500 RUN
```

Full Program: RML

```
  10 LET BX=14:LET A$=CHR$(195)+CHR$(195)+CHR$(195)+CHR
$(195):LET YD=1:LET SC=0:LET BALL=1
  20 DIMB(37,24)
  30 REM
  40 REM
  50 PUT 12
  60 REM
  70 FORR=4 TO 7:FORC=1 TO 33:LET B(C,R)=1:NEXT C:NEXT
R
  80 REM
  90 FORR=4 TO 7:FORC=1 TO 31:PLOT C*2,(21-R)*3,255:NEX
T C:NEXT R
 100 PLOT BX*2,6,A$
 110 LET Y=10:LET X=BX+2:LET XD=RND(1)/2+0.5:LET PAUSE=
30
 120 PLOT X*2,(21-Y)*3,207
 130 FORR=1 TO 1200:NEXT R
 140 PLOT 0,57,BALL+250
 150 PLOT X*2,(21-Y)*3,207
 160 FORR=1 TO PAUSE:NEXT R
 170 GOSUB 290
 180 PLOT X*2,(21-Y)*3,32
 190 IF B(X,Y)=1 THEN LET YD=-YD:LET B(X,Y)=0:LET SC=SC
+1:IF SC=124 THEN GOTO 50
 200 IF Y=4 THEN LET PAUSE=1
 210 IF X<2 OR X>31 THEN LET XD=-XD
 220 IF Y<2 THEN LET YD=-YD
 230 IF Y=18 AND X>=BX AND X<=BX+4 THEN LET YD=-YD:GOTO
 250
 240 LET BB=INT(X):IF Y=18 AND BB>BX-2 AND BB<BX+6 THEN
 LET XD=-XD*0.9:LET YD=-YD
```

```
250 IF Y>=20 THEN GOTO 390
260 LET X=X+XD
270 LET Y=Y+YD
280 GOTO 150
290 REM move bat routine
300 REM
310 PLOT BX*2,6,"     "
320 LET G$=GET$(2)
330 IF G$="Z" THEN LET BX=BX-2
340 IF G$="X" THEN LET BX=BX+2
350 IF BX<1 THEN LET BX=1
360 IF BX>27 THEN LET BX=27
370 PLOT BX*2,6,A$
380 RETURN
390 REM
400 LET BALL=BALL+1
410 REM
420 IF BALL>5 THEN GOTO 440
430 GOTO 110
440 REM
450 REM
460 REM
470 PRINT"SCORE=";SC;",NEW GAME ?";
480 INPUT D$
490 IF D$="N" THEN END
500 RUN
```

Mastermind

ZX81

This is a particularly frustrating version of the peg game. The computer chooses a random code of four characters and it is up to you to guess the code correctly. You know that the computer can choose from the first six letters of the alphabet but you don't know which letters, nor which order it has placed them in. The game is further complicated by the fact that the computer will only tell you if you have a letter in the right place. You get nothing if you have a correct letter in the wrong place. How many goes will it take you to crack the code?

Under five – brilliant
Five to nine – average
Ten and over – hopeless!

Player keys

First six letters of the alphabet
Q if you want to give up

The Program

First of all we set up dimensions for two arrays A and B. A will be the computer's hidden code and B will be your guess (lines 40–50). The computer then chooses four letters from the first six of the alphabet to put into A (lines 70–90).

```
10 FAST
20 CLS
30 RAND
40 DIM A(4)
50 DIM B(4)
60 LET Y=0
70 FOR R=1 TO 4
80 LET A(R)=INT(RND*6)+38
90 NEXT R
```

Then we come to the main program loop, which is where you have to sharpen your wits! The computer prints a ? in each of the four positions, one after the other, waits for your

guess for that place and prints it up. This is achieved by the FOR . . . NEXT loop in lines 100–160. Once it has got all four guesses from you, the computer then checks B with A to see how many of your pieces are in the correct place (lines 170–200). For each correct letter the computer will print ■ (lines 210–230) and, if you have all four correct, it will jump out at line 240 to congratulate you. The number of guesses you have had goes up by one (line 250), and if you have had twenty guesses you lose (line 260) – you're probably completely frustrated by the whole thing at this point!

```
100 FOR R=1 TO 4
110 PRINT AT Y,R*6;"?"
120 INPUT A$
130 IF A$="Q" THEN GOTO 280
140 LET B(R)=CODE A$
150 PRINT AT Y,R*6;CHR$ B(R)
160 NEXT R
170 LET C=0
180 FOR R=1 TO 4
190 IF A(R)=B(R) THEN LET C=C+1
200 NEXT R
210 FOR R=1 TO C
220 PRINT AT Y,R;"■"
230 NEXT R
240 IF C=4 THEN GOTO 320
250 LET Y=Y+1
260 IF Y>20 THEN GOTO 330
270 GOTO 100
```

If you decide that the computer's code is far too fiendish for you, you may give up, in which case it will reveal the elusive code for you (lines 280–300). On the other hand, if you do manage to crack the code, the computer will tell you at line 320 just how many guesses you took (see p. 28 for your rating) and, finally, you'll be offered another game (lines 330–360).

```
280 FOR R=1 TO 4
290 PRINT AT Y,R*6;CHR$ A(R)
300 NEXT R
310 GOTO 330
320 PRINT "VERY CLEVER-YOU TOOK ";Y;" GOES"
330 PRINT "PLAY AGAIN Y/N"
340 INPUT A$
350 IF A$="Y" THEN RUN
360 STOP
```

Suggestions

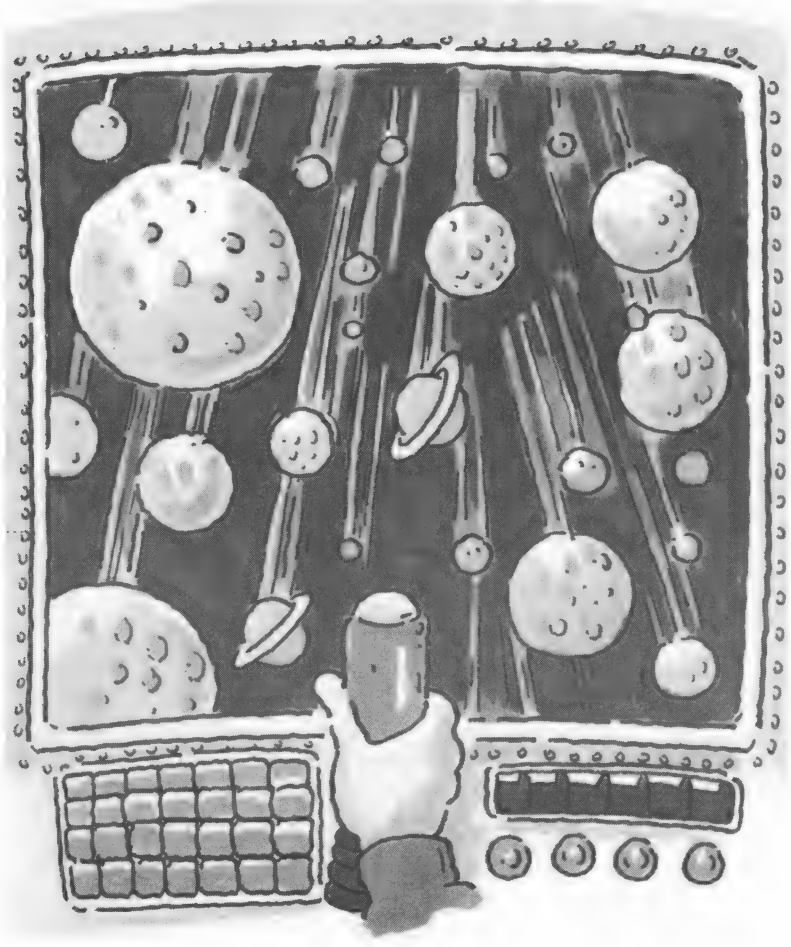
Make the game much harder by increasing the code to five characters chosen from seven or eight.

Full Program

```
 10 FAST
 20 CLS
 30 RAND
 40 DIM A(4)
 50 DIM B(4)
 60 LET Y=0
 70 FOR R=1 TO 4
 80 LET A(R)=INT(RND*6)+38
 90 NEXT R
100 FOR R=1 TO 4
110 PRINT AT Y,R*6;"?"
120 INPUT A$
130 IF A$="Q" THEN GOTO 280
140 LET B(R)=CODE A$
150 PRINT AT Y,R*6;CHR$ B(R)
160 NEXT R
170 LET C=0
180 FOR R=1 TO 4
190 IF A(R)=B(R) THEN LET C=C+1
200 NEXT R
210 FOR R=1 TO C
220 PRINT AT Y,R;"■"
230 NEXT R
240 IF C=4 THEN GOTO 320
250 LET Y=Y+1
260 IF Y>20 THEN GOTO 330
270 GOTO 100
280 FOR R=1 TO 4
290 PRINT AT Y,R*6;CHR$ A(R)
300 NEXT R
310 GOTO 330
320 PRINT "VERY CLEVER-YOU TOOK ";Y;" GOES"
330 PRINT "PLAY AGAIN Y/N"
340 INPUT A$
350 IF A$="Y" THEN RUN
360 STOP
```

Asteroid Belt

BBC, ZX Spectrum, RML



You are the captain of the Moon-shuttle *Orion*, on a routine trip with several passengers and full cargo-holds. Normally this short run to the moon is trouble-free, managed largely by the auto-pilot. However, on this trip, something goes badly wrong. The ship's defence systems fail to warn you that you are approaching a heavy asteroid storm. Only at the last moment do you realize the danger your ship is in, so that you just have time to switch over to manual control. There is no way to avoid this storm. The only hope for the survival of your ship and its passengers is for you to steer your way straight through the centre of the storm.

Player keys

Any key will make the ship change direction.

The Program

First of all, let's set some basic values for the computer to work from. X is the x-coordinate of your ship, Y the y-coordinate. Y remains constant. S is your score and D is the direction of your ship.

BBC

```
10 MODE 4
20 *FX11,0
30 LET X=15:LET Y=10:LET S=0:LET D=1
40 VDU23;8202;0;0;0;
```

Spectrum

```
10 BORDER 1:PAPER 0:INK 7:CLS
20 REM
30 LET X=15:LET Y=10:LET S=0:LET D=1
40 POKE 23692,255
```

RML

```
10 PUT 12
20 RANDOMIZE
30 LET X=30:LET Y=30:LET S=0:LET D=2
40 FORR=1 TO  25:PRINT:NEXT R
```

Next comes the main program loop, which begins by printing an asterisk at random on the screen (line 50), printing your ship and printing up your score so far (line 70). The computer checks whether you have got through the asteroid belt (line 90), checks whether you are pressing a key and, if so, changes the direction of your ship (line 100). Lines 120 and 130 prevent your ship going off the screen and line 140 checks whether you have crashed into an asteroid. The computer then returns to the beginning of the main loop at line 50.

BBC

```
 50 PRINT TAB(RND(30),31);"*"
 70 PRINT TAB(X,Y);"V":PRINTTAB(0,0);S
 80 SOUND1,-5,X,1
 90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":SOUND2,-5,125,20:GOTO 210
100 IF INKEY$(8)<>"" THEN LET D=-D
110 LET X=X+D
120 IF X<1 THEN LET X=1:LET D=-D
130 IF X>30 THEN LET X=30:LET D=-D
140 IF POINT(X*32+16,656)>0 THEN GOTO 160
150 GOTO 50
```

Spectrum

```
 50 PRINT AT 21,RND*29+1;"*"
 70 PRINT AT Y,X;"V":PRINT AT 0,0;S
 80 BEEP 0.01,X*2
 90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":BEEP 2,15:GOTO 210
100 IF INKEY$<>"" THEN LET D=-D
110 LET X=X+D
120 IF X<1 THEN LET X=1:LET D=-D
130 IF X>30 THEN LET X=30:LET D=-D
140 IF SCREEN$(Y+1,X)<>" " THEN GOTO 160
150 GOTO 50
```

RML

```
   50 PRINT TAB(INT(RND(1)*29)+1);"*"
   70 PLOT X,Y,"V"
   80 REM
   90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":GOTO 210
  100 IF GET$(8)<>"" THEN LET D=-D
  110 LET X=X+D
  120 IF X<2 THEN LET X=2:LET D=-D
  130 IF X>60 THEN LET X=60:LET D=-D
  140 IF POINT X,Y-3 >32 THEN GOTO 160
  150 GOTO 50
```

If you do have the misfortune to crash into an asteroid, line 140 sends you to this rather interesting crash routine.

BBC

```
  160 FOR R=1 TO 60
  170 SOUND4,-5,RND(20),1
  180 PRINT TAB(RND(5)+X-2,RND(5)+Y-2);CHR$(RND(14)+32)
  190 NEXT R
  200 PRINT TAB(0,21);"HA! HA! SOME PILOT YOU ARE!"
```

Spectrum

```
  160 FOR R=1 TO 60
  170 INK RND*6+1
  180 PRINT AT RND*4+Y-2,RND*4+X-2;CHR$(RND*15+128)
  190 NEXT R
  200 PRINT AT 21,0;"HA! HA! SOME PILOT YOU ARE!"
```

RML

```
  160 FOR R=1 TO 60
  170 REM
  180 PLOT X,Y-3,195+R
  190 NEXT R
  200 PLOT 0,0,"HA! HA! SOME PILOT YOU ARE!"
```

Finally, you get offered a new game.

BBC

```
210 INPUT"PLAY AGAIN? (Y/N)"A$
220 IF A$="Y" OR A$="y" THEN RUN
230 *FX11,55
240 STOP
```

Spectrum

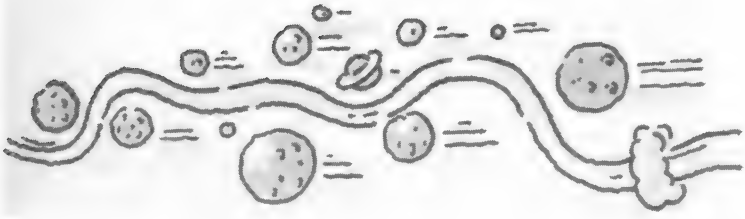
```
210 INPUT"PLAY AGAIN? (Y/N)";A$
220 IF A$="Y" OR A$="y" THEN RUN
230 REM
240 STOP
```

RML

```
210 INPUT"PLAY AGAIN? (Y/N)";A$
220 IF A$="Y" OR A$="y" THEN RUN
230 REM
240 STOP
```

Suggestions

Introduce a variable to count how many asteroids you miss and then get your score printed up as part of the crash routine.



Full Program: BBC

```
   10 MODE 4
   20 *FX11,0
   30 LET X=15:LET Y=10:LET S=0:LET D=1
   40 VDU23;8202;0;0;0;
   50 PRINT TAB(RND(30),31);"*"
   70 PRINT TAB(X,Y);"V":PRINTTAB(0,0);S
   80 SOUND1,-5,X,1
   90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":SOUND2,-5,125,20:GOTO 210
  100 IF INKEY$(8)<>"" THEN LET D=-D
  110 LET X=X+D
  120 IF X<1 THEN LET X=1:LET D=-D
  130 IF X>30 THEN LET X=30:LET D=-D
  140 IF POINT(X*32+16,656)>0 THEN GOTO 160
  150 GOTO 50
  160 FOR R=1 TO 60
  170 SOUND4,-10,RND(20),1
  180 PRINT TAB(RND(5)+X-2,RND(5)+Y-2);CHR$(RND(14)+32)
  190 NEXT R
  200 PRINT TAB(0,21);"HA! HA! SOME PILOT YOU ARE!"
  210 INPUT"PLAY AGAIN? (Y/N)"A$
  220 IF A$="Y" OR A$="y" THEN RUN
  230 *FX11,55
  240 STOP
```

Full Program: ZX Spectrum

```
   10 BORDER 1:PAPER 0:INK 7:CLS
   20 REM
   30 LET X=15:LET Y=10:LET S=0:LET D=1
   40 POKE 23692,255
   50 PRINT AT 21,RND*29+1;"*"
   70 PRINT AT Y,X;"V":PRINT AT 0,0;S
   80 BEEP 0.01,X*2
   90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":BEEP 2,15:GOTO 210
  100 IF INKEY$<>"" THEN LET D=-D
  110 LET X=X+D
  120 IF X<1 THEN LET X=1:LET D=-D
  130 IF X>30 THEN LET X=30:LET D=-D
  140 IF SCREEN$(Y+1,X)<>" " THEN GOTO 160
  150 GOTO 50
  160 FOR R=1 TO 60
  170 INK RND*6+1
  180 PRINT AT RND*4+Y-2,RND*4+X-2;CHR$(RND*15+128)
  190 NEXT R
  200 PRINT AT 21,0;"HA! HA! SOME PILOT YOU ARE!"
```

```
  210 INPUT"PLAY AGAIN? (Y/N)";A$
  220 IF A$="Y" OR A$="y" THEN RUN
  230 REM
  240 STOP
```

Full Program: RML

```
   10 PUT 12
   20 RANDOMIZE
   30 LET X=30:LET Y=30:LET S=0:LET D=2
   40 FORR=1 TO  25:PRINT:NEXT R
   50 PRINT TAB(INT(RND(1)*29)+1);"*"
   70 PLOT X,Y,"V"
   80 REM
   90 LET S=S+1:IF S>254 THEN PRINT"WELL DONE-MISSION CO
MPLETE":GOTO 210
  100 IF GET$(8)<>"" THEN LET D=-D
  110 LET X=X+D
  120 IF X<2 THEN LET X=2:LET D=-D
  130 IF X>60 THEN LET X=60:LET D=-D
  140 IF POINT X,Y-3 >32 THEN GOTO 160
  150 GOTO 50
  160 FOR R=1 TO 60
  170 REM
  180 PLOT X,Y-3,195+R
  190 NEXT R
  200 PLOT 0,0,"HA! HA! SOME PILOT YOU ARE!"
  210 INPUT"PLAY AGAIN? (Y/N)";A$
  220 IF A$="Y" OR A$="y" THEN RUN
  230 REM
  240 STOP
```

Danger Orbit

BBC, ZX Spectrum, RML, ZX81

Trapped into an ever-decreasing orbit round a ruined alien city, your only hope of landing safely is to try to flatten the buildings beneath you and make room for your ship to land. Unfortunately, your descent is so rapid that you are in danger of crashing into the tallest ruins unless you flatten them first.

Player keys

Any key will fire a bomb.

The Program

First of all we ask for an array of forty buildings (line 30) of random height (line 60). The RML micro uses a grey block to construct the buildings, the Spectrum uses graphic shift 8, while the other two use the letter H (line 70, *70–75 ZX81*).

BBC

```
10 MODE4
20 VDU23;8202;0;0;0;
30 DIMB(40)
40 COLOUR1
50 FOR X=1 TO 31
60 LET Y=RND(12):LET B(X+1)=Y
70 FORR=21 TO 21-Y STEP -1:PRINTTAB(X,R);"H"
80 NEXT R
90 NEXT X
```

Spectrum

```
10 REM
20 RANDOMIZE
30 DIM B(40)
40 INK 2:BORDER 4:PAPER 7:CLS
50 FOR X=1 TO 31
60 LET Y=INT(RND*14):LET B(X+1)=Y
70 FORR=21 TO 21-Y STEP -1:PRINT AT R,X;"■"
80 NEXT R
90 NEXT X
```

RML

```
 5 GRAPH
10 PUT 12
20 REM
30 DIM B(40)
40 REM
50 FOR X=1 TO 31
60 Y=INT(RND(1)*12)+1:LET B(X+1)=Y
70 FORR=21 TO 21-Y STEP -1:PLOT X*2,(21-R)*3,191
80 NEXT R
90 NEXT X
```

ZX81

```
10 FAST
20 RAND
30 DIMB(40)
40 CLS
50 FOR X=1 TO 31
60 LET Y=RND(12)
65 LET B(X+1)=Y
70 FORR=21 TO 21-Y STEP -1
75 PRINT AT R,X;"H"
80 NEXT R
90 NEXT X
```

Just before we begin the main program, we will need to set initial values for the ship's coordinates and the score. We also set a value for F, which will mean that only one bomb can fall at a time. F = 0 when there is no bomb, and F = 1 when one is on the screen.

BBC and Spectrum

```
100 LET PX=0:LET PY=1:LET Score=0:LET F=0
```

RML and ZX81

```
100 LET PX=0:LET PY=1:LET SC=0:LET F=0
```

Now we are ready for the main loop, which prints the ship (line 110), prints the bomb if there is one (line 120) and checks to see whether you have pressed a key to fire (line 130). Then it will unprint the bomb (line 140), unprint your ship (line 150) and change its coordinates (line 160). Finally, the computer checks to see whether you have landed successfully (line 170) or whether you have crashed into a building (line 180).

Then the loop begins again.

BBC

```
  110 PRINTTAB(PX,PY);">"
  120 IF F=1 THEN GOSUB 200
  130 LET G$=INKEY$(5):IF G$<>"" AND F=0 THEN LET F=1:LE
T BY=PY+1:LET BX=PX
  140 IF F=1 THEN GOSUB 220
  150 PRINTTAB(PX,PY);" "
  160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
  170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
  180 IF PY>=21-B(PX+1) THEN GOTO 320
  190 GOTO 110
```

Spectrum

```
  110 PRINT AT PY,PX;">"
  120 IF F=1 THEN GOSUB 200
  130 LET G$=INKEY$:IF G$<>"" AND F=0 THEN LET F=1:LET B
Y=PY+1:LET BX=PX
  140 IF F=1 THEN GOSUB 220
  150 PRINT AT PY,PX;" "
  160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
  170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
  180 IF PY>=21-B(PX+1) THEN GOTO 320
  190 GOTO 110
```

RML

```
  110 PLOT PX*2,(21-PY)*3,">"
  120 IF F=1 THEN GOSUB 200
  130 LET G$=GET$(5):IF G$<>"" AND F=0 THEN LET F=1:LET
BY=PY+1:LET BX=PX
  140 IF F=1 THEN GOSUB 220
  150 PLOT PX*2,(21-PY)*3,32
  160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
```

```
170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
180 IF PY>=21-B(PX+1) THEN GOTO 320
190 GOTO 110
```

ZX81

```
110 PRINT AT PY,PX;">"
120 IF F=1 THEN GOSUB 200
130 IF INKEY$="" OR F=0 THEN GOTO 140
132 LET F=1
134 LET BY=PY+1
136 LET BX=PX
140 IF F=1 THEN GOSUB 220
150 PRINT AT PY,PX;" "
160 LET PX=PX+1
162 IF PX<32 THEN GOTO 170
165 LET PX=0
167 LET PY=PY+1
170 IF PY<21 THEN GOTO 180
172 PRINT"WELL DONE"
174 GOTO 350
180 IF PY>=21-B(PX+1) THEN GOTO 320
190 GOTO 110
```

Once the computer has checked to see whether there is a bomb on the screen (line 120), it jumps to a small subroutine to print the bomb and then returns to the main loop.

BBC

```
200 PRINTTAB(BX,BY);"*"
210 RETURN
```

Spectrum and ZX81

```
200 PRINT AT BY,BX;"*"
210 RETURN
```

RML

```
200 PLOT BX*2,(21-BY)*3,"*"
210 RETURN
```

At line 140, the computer checks again to see whether there is a bomb on the screen and jumps to another small subroutine to unprint it. The coordinates of the bomb are then changed (line 240) and the computer checks to see whether a building has been hit (line 250) before returning to the main loop.

BBC

```
220 PRINTTAB(BX,BY);" "
230 SOUND2,-7,(21-BY),1
240 LET BY=BY+1
250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
260 RETURN
```

Spectrum

```
220 PRINT AT BY,BX;" "
230 BEEP 0.1,(21-BY)/2
240 LET BY=BY+1
250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
260 RETURN
```

RML

```
220 PLOT BX*2,(21-BY)*3,32
230 REM
240 LET BY=BY+1
250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
260 RETURN
```

ZX81

```
220 PRINT AT BY,BX;" "
230 REM
240 LET BY=BY+1
250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
260 RETURN
```

If your bomb has hit a ruined building, the computer will jump to a further subroutine at line 270, which unprints the building and adds the height of the building to your score before returning to the main loop.

BBC

```
  270 FOR R=21-B(BX+1) TO 21:PRINTTAB(BX,R);" ":NEXT R
  280 LET Score=Score+B(BX+1):PRINTTAB(0,0);"SCORE ";Sco
re
  290 LET F=0
  300 LET B(BX+1)=0
  310 RETURN
```

Spectrum

```
  270 FOR R=21-B(BX+1) TO 21:PRINT AT R,BX;" ":NEXT R
  280 LET SC=SC+1:PRINT AT 0,0;"Score=";SC
  290 LET F=0
  300 LET B(BX+1)=0
  310 RETURN
```

RML

```
  270 FOR R=21-B(BX+1) TO 21:PLOT BX*2,(21-R)*3,32:NEXT
R
  280 LET SC=SC+B(BX+1):PUT 12:PRINT"SCORE=";SC
  290 LET F=0
  300 LET B(BX+1)=0
  310 RETURN
```

ZX81

```
  270 FOR R=21-B(BX+1) TO 21
  275 PRINT AT R,BX;" "
  278 NEXT R
  280 LET SC=SC+B(BX+1)
  285 PRINT AT 0,0;"SCORE ";SC
  290 LET F=0
  300 LET B(BX+1)=0
  310 RETURN
```

So what happens when you do hit a ruined building, as you certainly will until you become very expert? Here is a special explosion effect.

BBC

```
320 FOR R=1 TO 200
330 FOR S=1 TO 6:VDU19,0,R,0,0,0:NEXT S:NEXT R
340 MODE 4
```

Spectrum

```
320 FOR R=1 TO 10
330 FOR S=1 TO 2:PAPER S:CLS:NEXT S:NEXT R
340 CLS
```

RML

```
320 FOR R=1 TO 60
330 PLOT PX*2,(21-PY)*3,195+R:NEXT R
340 REM
```

ZX81

```
320 FOR R=1 TO 200
330 PRINT AT PY,PX;CHR$(129+RND*10)
340 NEXT R
```

Finally, the computer will offer you another game, wait for your answer and respond. If you press Y for 'yes', the program will run again from the beginning, while pressing N for 'no' will end the program and leave the computer waiting for your next command.

BBC, Spectrum and ZX81

```
350 PRINT"PLAY AGAIN (Y/N) ";
360 INPUT G$
370 IF G$="Y" THEN RUN
380 STOP
```

RML

```
350 PRINT"PLAY AGAIN (Y/N) ";
360 INPUT G$
370 IF G$="Y" THEN RUN
380 TEXT:PUT 12:END
```

Suggestions

If you find this game a bit hard, or a bit easy, try varying the height of the buildings (line 60). You could also make it possible to input the level of difficulty.

Full Program: BBC

```
  10 MODE4
  20 VDU23;8202;0;0;0;
  30 DIMB(40)
  40 COLOUR1
  50 FOR X=1 TO 31
  60 LET Y=RND(12):LET B(X+1)=Y
  70 FORR=21 TO 21-Y STEP -1:PRINTTAB(X,R);"H"
  80 NEXT R
  90 NEXT X
 100 LET PX=0:LET PY=1:LET Score=0:LET F=0
 110 PRINTTAB(PX,PY);">"
 120 IF F=1 THEN GOSUB 200
 130 LET G$=INKEY$(5):IF G$<>"" AND F=0 THEN LET F=1:LE
T BY=PY+1:LET BX=PX
 140 IF F=1 THEN GOSUB 220
 150 PRINTTAB(PX,PY);" "
 160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
 170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
 180 IF PY>=21-B(PX+1) THEN GOTO 320
 190 GOTO 110
 200 PRINTTAB(BX,BY);"*"
 210 RETURN
 220 PRINTTAB(BX,BY);" "
 230 SOUND2,-7,(21-BY),1
 240 LET BY=BY+1
 250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
 260 RETURN
 270 FOR R=21-B(BX+1) TO 21:PRINTTAB(BX,R);" ":NEXT R
 280 LET Score=Score+B(BX+1):PRINTTAB(0,0);"SCORE ";Sco
re
 290 LET F=0
 300 LET B(BX+1)=0
 310 RETURN
 320 FOR R=1 TO 200
 330 FOR S=1 TO 6:VDU19,0,R,0,0,0:NEXT S:NEXT R
 340 MODE 4
 350 PRINT"PLAY AGAIN (Y/N) ";
 360 INPUT G$
 370 IF G$="Y" THEN RUN
 380 STOP
```

Full Program: Spectrum

```
  10 REM
  20 RANDOMIZE
  30 DIM B(40)
  40 INK 2:BORDER 4:PAPER 7:CLS
  50 FOR X=1 TO 31
  60 LET Y=INT(RND*14):LET B(X+1)=Y
  70 FORR=21 TO 21-Y STEP -1:PRINT AT R,X;"■"
  80 NEXT R
  90 NEXT X
 100 LET PX=0:LET PY=1:LET SC=0:LET F=0
 110 PRINT AT PY,PX;">"
 120 IF F=1 THEN GOSUB 200
 130 LET G$=INKEY$:IF G$<>"" AND F=0 THEN LET F=1:LET B
Y=PY+1:LET BX=PX
 140 IF F=1 THEN GOSUB 220
 150 PRINT AT PY,PX;" "
 160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
 170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
 180 IF PY>=21-B(PX+1) THEN GOTO 320
 190 GOTO 110
 200 PRINT AT BY,BX;"*"
 210 RETURN
 220 PRINT AT BY,BX;" "
 230 BEEP 0.1,(21-BY)/2
 240 LET BY=BY+1
 250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
 260 RETURN
 270 FOR R=21-B(BX+1) TO 21:PRINT AT R,BX;" ":NEXT R
 280 LET SC=SC+1:PRINT AT 0,0;"Score=";SC
 290 LET F=0
 300 LET B(BX+1)=0
 310 RETURN
 320 FOR R=1 TO 10
 330 FOR S=1 TO 2:PAPER S:CLS:NEXT S:NEXT R
 340 CLS
 350 PRINT"PLAY AGAIN (Y/N) ";
 360 INPUT G$
 370 IF G$="Y" THEN RUN
 380 STOP
```

Full Program: RML

```
    5 GRAPH
   10 PUT 12
   20 REM
   30 DIM B(40)
   40 REM
   50 FOR X=1 TO 31
   60 Y=INT(RND(1)*12)+1:LET B(X+1)=Y
   70 FORR=21 TO 21-Y STEP -1:PLOT X*2,(21-R)*3,191
   80 NEXT R
   90 NEXT X
  100 LET PX=0:LET PY=1:LET SC=0:LET F=0
  110 PLOT PX*2,(21-PY)*3,">"
  120 IF F=1 THEN GOSUB 200
  130 LET G$=GET$(5):IF G$<>"" AND F=0 THEN LET F=1:LET
BY=PY+1:LET BX=PX
  140 IF F=1 THEN GOSUB 220
  150 PLOT PX*2,(21-PY)*3,32
  160 LET PX=PX+1:IF PX>31 THEN LET PX=0:LET PY=PY+1
  170 IF PY>20 THEN PRINT"WELL DONE":GOTO 350
  180 IF PY>=21-B(PX+1) THEN GOTO 320
  190 GOTO 110
  200 PLOT BX*2,(21-BY)*3,"*"
  210 RETURN
  220 PLOT BX*2,(21-BY)*3,32
  230 REM
  240 LET BY=BY+1
  250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
  260 RETURN
  270 FOR R=21-B(BX+1) TO 21:PLOT BX*2,(21-R)*3,32:NEXT
R
  280 LET SC=SC+B(BX+1):PUT 12:PRINT"SCORE=";SC
  290 LET F=0
  300 LET B(BX+1)=0
  310 RETURN
  320 FOR R=1 TO 60
  330 PLOT PX*2,(21-PY)*3,195+R:NEXT R
  340 REM
  350 PRINT"PLAY AGAIN (Y/N) ";
  360 INPUT G$
  370 IF G$="Y" THEN RUN
  380 TEXT:PUT 12:END
```

Full Program: ZX81

```
 10 FAST
 20 RAND
 30 DIMB(40)
 40 CLS
 50 FOR X=1 TO 31
 60 LET Y=RND(12)
 65 LET B(X+1)=Y
 70 FORR=21 TO 21-Y STEP -1
 75 PRINT AT R,X;"H"
 80 NEXT R
 90 NEXT X
100 LET PX=0:LET PY=1:LET SC=0:LET F=0
110 PRINT AT PY,PX;">"
120 IF F=1 THEN GOSUB 200
130 IF INKEY$="" OR F=0 THEN GOTO 140
132 LET F=1
134 LET BY=PY+1
136 LET BX=PX
140 IF F=1 THEN GOSUB 220
150 PRINT AT PY,PX;" "
160 LET PX=PX+1
162 IF PX<32 THEN GOTO 170
165 LET PX=0
167 LET PY=PY+1
170 IF PY<21 THEN GOTO 180
172 PRINT"WELL DONE"
174 GOTO 350
180 IF PY>=21-B(PX+1) THEN GOTO 320
190 GOTO 110
200 PRINT AT BY,BX;"*"
210 RETURN
220 PRINT AT BY,BX;" "
230 REM
240 LET BY=BY+1
250 IF BY>=21-B(BX+1)+1 THEN GOTO 270
260 RETURN
270 FOR R=21-B(BX+1) TO 21
275 PRINT AT R,BX;" "
278 NEXT R
280 LET SC=SC+B(BX+1)
285 PRINT AT 0,0;"SCORE ";SC
290 LET F=0
300 LET B(BX+1)=0
310 RETURN
320 FOR R=1 TO 200
330 PRINT AT PY,PX;CHR$(129+RND*10)
340 NEXT R
350 PRINT"PLAY AGAIN (Y/N) ";
360 INPUT G$
370 IF G$="Y" THEN RUN
380 STOP
```

One-way Street

BBC, ZX Spectrum, RML

Driving home late one night, you accidentally make a terrible mistake and turn into a narrow, twisting, one-way street from the wrong end. Suddenly, to your horror, your brakes go, making it impossible for you to stop or turn your car round. Your only hope is to steer your way through the oncoming traffic, trying to avoid a crash!

Player keys

z – left
x – right

The Program

As usual, we have to give the computer some basic information to start with. P1 and P2 are the initial x-coordinates of the street, Q is the direction of the street, X is the position of your car and E is your score (line 30).

BBC

```
10 MODE4
20 VDU23,255,231,231,231,231,231,231,231,231
30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
40 *FX11,8
50 *FX12,6
60 VDU23;8202;0;0;0;
```

Spectrum

```
10 BORDER 1
20 REM
30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
40 REM
50 REM
60 POKE 23692,255
```

RML

```
10 PUT 12
20 REM
30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
40 RANDOMIZE
50 REM
60 REM
```

There are, in effect, two main loops to this program, one set into the other. The inner loop (lines 70–180) is a FOR . . . NEXT routine, which prints the street (line 80), prints your car (line 120), checks to see if you are pressing the player keys (line 120), moves your car accordingly, and finally checks to see whether you have crashed (line 170). The ZX Spectrum street uses graphic shift 5 and 2 for P1 and the inverse for P2.

The outer loop prints the oncoming cars (line 190), reads Q to see whether the course should change direction, and then sends the computer back to line 70 to begin the main loop again.

BBC

```
   70 FOR S=1 TO 10
   80 PRINT TAB(ABS(P1),30);CHR$255;CHR$255;TAB(P2,30);C
HR$255;CHR$255
   90 PRINT
  100 LET P1=P1+Q:LET P2=P1+8
  110 SOUND1,-5,X,1
  120 PRINT TAB(X,11);"H"
  130 REM
  140 LET G$=INKEY$(5)
  150 IF G$="Z" THEN LET X=X-1
  160 IF G$="X" THEN LET X=X+1
  170 IF POINT(X*32+8,624)<>0 THEN GOTO 250
  180 NEXT S
  190 PRINT TAB(P1+1+RND(5),30);"#"
  200 SOUND2,-5,125,2
  210 READ Q:IF Q=999 THEN GOTO 240
  220 GOTO 70
```

Spectrum

```
 70 FOR S=1 TO 10
 80 PRINT AT 21,P1;"▗▄▖";AT 21,P2;"▝▀▘"
 90 PRINT
100 LET P1=P1+Q:LET P2=P1+8
110 BEEP 0.01,X
120 PRINT AT 11,X;"H"
130 REM
```

```
 140 LET G$=INKEY$
 150 IF G$="Z" THEN LET X=X-1
 160 IF G$="X" THEN LET X=X+1
 170 IF SCREEN$(11,X)<>" " THEN GOTO 250
 180 NEXT S
 190 PRINT AT 20,P1+2+RND*5;"#"
 200 BEEP 0.01,40
 210 READ Q:IF Q=999 THEN GOTO 240
 215 POKE 23692,255
 220 GOTO 70
```

RML

```
  70 FOR S=1 TO 10
  80 PRINT TAB(ABS(P1));CHR$(213);CHR$(213);TAB(P2);CHR
$(213);CHR$(213)
  90 REM
 100 LET P1=P1+Q:LET P2=P1+8
 110 REM
 120 PLOT X*2,13*3,"H"
 130 REM
 140 G$=GET$(2)
 150 IF G$="Z" THEN X=X-1
 160 IF G$="X" THEN X=X+1
 170 IF POINT(X*2,36)=2 OR POINT(X*2,36)>32 THEN GOTO 2
50
 180 NEXT S
 190 PLOT (P1+INT(RND(1)*4)+2)*2-10*Q,0,65
 200 REM
 210 READ Q:IF Q=999 THEN GOTO 240
 220 GOTO 70
```

Every ten steps, the street will change direction as the value of Q changes. So we must give the computer some values of Q to read. RML users will have to fit the last two numbers on to line 235, as the computer cannot hold all these characters on one line.

BBC, Spectrum and RML

```
 230 DATA 0,0,0,1,0,0,-1,-1,0,0,1,1,0,-0.2,-0.5,-0.5,-0
.5,1,0,0,0,-0.5,0.5,-1,0,1,0,-1,-0.3,1,0.5,0.5,-0.7,-1,0
,1,0.7,0,0,-1,-1,-0.5,999
```

Back in the main loop, line 170 checks whether your car has hit anything (another car, or the edge of the street) and, if so, jumps to line 250, which will offer you another game. Alternatively, if your reactions have been very quick and you have got through the street unharmed, line 210 reads Q, realizes that you have reached the end of the course and jumps to line 240 for the win routine.

BBC

```
240 PRINT TAB(0,0);"CONGRATULATIONS!-COURSE COMPLETE"
250 INPUT"PLAY AGAIN? Y/N "A$
260 IF A$="Y" THEN RUN
270 *FX11,55
280 STOP
```

Spectrum

```
240 PRINT AT 0,0;"CONGRATULATIONS!-COURSE COMPLETE"
250 INPUT"PLAY AGAIN? Y/N ";A$
260 IF A$="Y" THEN RUN
270 REM
280 STOP
```

RML

```
235 DATA -0.5,999
240 PRINT"CONGRATULATIONS!-COURSE COMPLETE"
250 INPUT"PLAY AGAIN? Y/N ";A$
260 IF A$="Y" THEN RUN
270 REM
280 END
```

Suggestions

Try creating a really spectacular explosion effect between lines 240 and 250.

You could also get the computer to calculate a score for you in the main loop, perhaps by giving each of the oncoming cars a value, or by adding one for each completed loop. You could then add a line to your crash routine, making the computer print your score for you.

Full Program: BBC

```
   10 MODE4
   20 VDU23,255,231,231,231,231,231,231,231,231
   30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
   40 *FX11,8
   50 *FX12,6
   60 VDU23;8202;0;0;0;
   70 FOR S=1 TO 10
   80 PRINT TAB(ABS(P1),30);CHR$255;CHR$255;TAB(P2,30);C
HR$255;CHR$255
   90 PRINT
  100 LET P1=P1+Q:LET P2=P1+8
  110 SOUND1,-5,X,1
  120 PRINT TAB(X,11);"H"
  130 REM
  140 LET G$=INKEY$(5)
  150 IF G$="Z" THEN LET X=X-1
  160 IF G$="X" THEN LET X=X+1
  170 IF POINT(X*32+8,624)<>0 THEN GOTO 250
  180 NEXT S
  190 PRINT TAB(P1+1+RND(5),30);"#"
  200 SOUND2,-5,125,2
  210 READ Q:IF Q=999 THEN GOTO 240
  220 GOTO 70
  230 DATA 0,0,0,1,0,0,-1,-1,0,0,1,1,0,-0.2,-0.5,-0.5,-0
.5,1,0,0,0,-0.5,0.5,-1,0,1,0,-1,-0.3,1,0.5,0.5,-0.7,-1,0
,1,0.7,0,0,-1,-1,-0.5,999
  240 PRINT TAB(0,0);"CONGRATULATIONS!-COURSE COMPLETE"
  250 INPUT"PLAY AGAIN? Y/N "A$
  260 IF A$="Y" THEN RUN
  270 *FX11,55
  280 STOP
```

Full Program: ZX Spectrum

```
   10 BORDER 1
   20 REM
   30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
   40 REM
   50 REM
   60 POKE 23692,255
   70 FOR S=1 TO 10
   80 PRINT AT 21,P1;"■";AT 21,P2;"■"
   90 PRINT
  100 LET P1=P1+Q:LET P2=P1+8
  110 BEEP 0.01,X
  120 PRINT AT 11,X;"H"
  130 REM
  140 LET G$=INKEY$
  150 IF G$="Z" THEN LET X=X-1
```

```
  160 IF G$="X" THEN LET X=X+1
  170 IF SCREEN$(11,X)<>" " THEN GOTO 250
  180 NEXT S
  190 PRINT AT 20,P1+2+RND*5;"#"
  200 BEEP 0.01,40
  210 READ Q:IF Q=999 THEN GOTO 240
  215 POKE 23692,255
  220 GOTO 70
  230 DATA 0,0,0,1,0,0,-1,-1,0,0,1,1,0,-0.2,-0.5,-0.5,-0
.5,1,0,0,0,-0.5,0.5,-1,0,1,0,-1,-0.3,1,0.5,0.5,-0.7,-1,0
,1,0.7,0,0,-1,-1,-0.5,999
  240 PRINT AT 0,0;"CONGRATULATIONS!-COURSE COMPLETE"
  250 INPUT"PLAY AGAIN? Y/N ";A$
  260 IF A$="Y" THEN RUN
  270 REM
  280 STOP
```

Full Program: RML

```
   10 PUT 12
   20 REM
   30 LET P1=11:LET P2=P1+8:LET Q=0:LET X=15
   40 RANDOMIZE
   50 REM
   60 REM
   70 FOR S=1 TO 10
   80 PRINT TAB(ABS(P1));CHR$(213);CHR$(213);TAB(P2);CHR
$(213);CHR$(213)
   90 REM
  100 LET P1=P1+Q:LET P2=P1+8
  110 REM
  120 PLOT X*2,13*3,"H"
  130 REM
  140 G$=GET$(2)
  150 IF G$="Z" THEN X=X-1
  160 IF G$="X" THEN X=X+1
  170 IF POINT(X*2,36)=2 OR POINT(X*2,36)>32 THEN GOTO 2
50
  180 NEXT S
  190 PLOT (P1+INT(RND(1)*4)+2)*2-10*Q,0,65
  200 REM
  210 READ Q:IF Q=999 THEN GOTO 240
  220 GOTO 70
  230 DATA 0,0,0,1,0,0,-1,-1,0,0,1,1,0,-0.2,-0.5,-0.5,-0
.5,1,0,0,0,-0.5,0.5,-1,0,1,0,-1,-0.3,1,0.5,0.5,-0.7,-1,0
,1,0.7,0,0,-1,-1
  235 DATA -0.5,999
  240 PRINT"CONGRATULATIONS!-COURSE COMPLETE"
  250 INPUT"PLAY AGAIN? Y/N ";A$
  260 IF A$="Y" THEN RUN
  270 REM
  280 END
```

Memory Mix

ZX81

An infuriating game, designed to test your powers of observation. The computer will print up ten random letters in a diagonal down the screen. You get a short time to look at the letters before they disappear. The computer then prints them up again, but this time there are only nine letters. One letter is missing, but which one? Sounds easy, doesn't it? Try it and see!

The Program

We start by setting up two arrays, A and B (lines 20–30). A will hold the first twenty letters of the alphabet, while B will hold the ten the computer picks for you to see. Lines 40–60 put the twenty letters into A and lines 70–90 choose ten of these at random to put into B.

```
10 SLOW
20 DIM A(20)
30 DIM B(10)
40 FOR R=1 TO 20
50 LET A(R)=R+37
60 NEXT R
70 FOR R=1 TO 10
80 LET B(R)=A(INT(RND*20)+1)
90 NEXT R
```

The computer then prints up the ten letters in a diagonal line down the screen (lines 110–130), chooses which letter it is going to take away (line 140) and gives it the value M (line 150), before taking it away (line 160).

```
100 CLS
110 FOR R=1 TO 10
120 PRINT AT R,R*2;CHR$ B(R)
130 NEXT R
140 LET W=INT(RND*10)+1
150 LET M=B(W)
160 LET B(W)=0
```

Memory Mix consists of a series of small loops, rather than one main loop. This one prints a line along the bottom of the screen, so that you can see your time ticking away:

```
170 FOR R=1 TO 126
180 PLOT R/2,1
190 NEXT R
```

Then comes the moment of truth. The computer prints up the same array of letters, but this time one is missing (lines 210–230). Can you spot which one it is? The computer waits for your guess at line 240 and will tell you whether you are right (line 340) or wrong (line 290). If you are wrong, it will tell you which letter was missing (line 310). Finally, you are offered another game (lines 360–390).

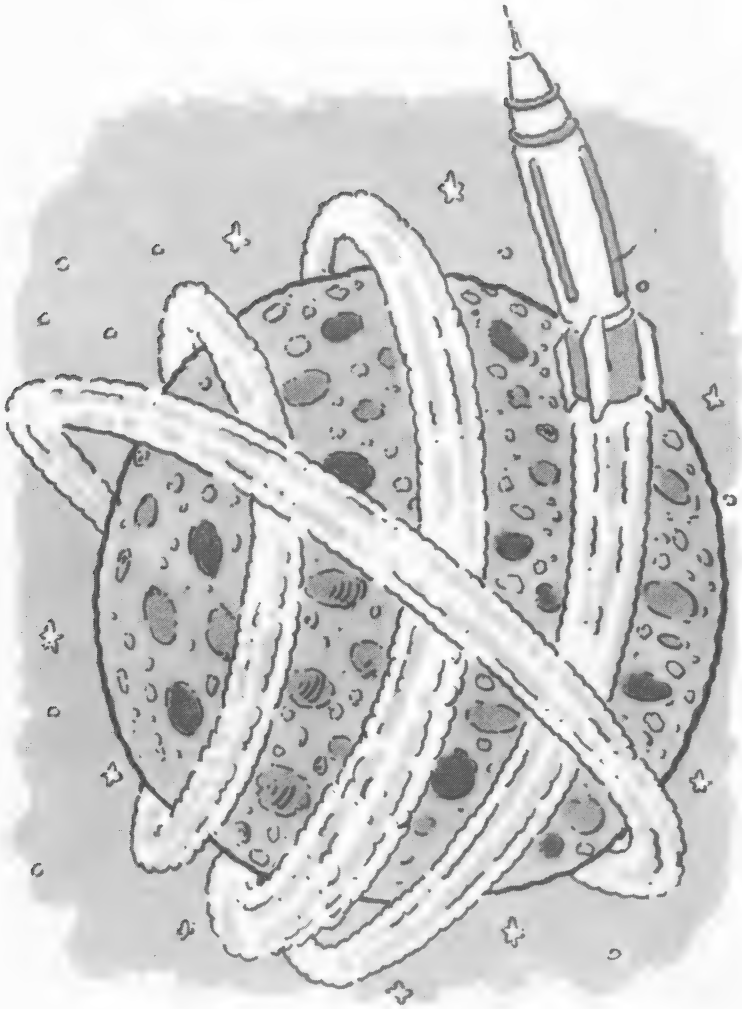
```
200 CLS
210 FOR R=1 TO 10
220 PRINT AT R,R*2;CHR$ B(R)
230 NEXT R
240 PRINT "WHAT IS THE MISSING LETTER?"
250 INPUT C$
260 IF CODE (C$)=M THEN GOTO 330
270 PRINT
280 PRINT
290 PRINT "WRONG -- THE MISSING LETTER"
300 PRINT
310 PRINT "            IS -";CHR$ M;"-"
320 GOTO 350
330 PRINT
340 PRINT "*****CORRECT*****"
350 PRINT
360 PRINT "PLAY AGAIN Y/N"
370 INPUT Z$
380 IF Z$="Y" THEN RUN
390 STOP
```

Full Program

```
 10 SLOW
 20 DIM A(20)
 30 DIM B(10)
 40 FOR R=1 TO 20
 50 LET A(R)=R+37
 60 NEXT R
 70 FOR R=1 TO 10
 80 LET B(R)=A(INT(RND*20)+1)
 90 NEXT R
100 CLS
110 FOR R=1 TO 10
120 PRINT AT R,R*2;CHR$ B(R)
130 NEXT R
140 LET W=INT(RND*10)+1
150 LET M=B(W)
160 LET B(W)=0
170 FOR R=1 TO 126
180 PLOT R/2,1
190 NEXT R
200 CLS
210 FOR R=1 TO 10
220 PRINT AT R,R*2;CHR$ B(R)
230 NEXT R
240 PRINT "WHAT IS THE MISSING LETTER?"
250 INPUT C$
260 IF CODE (C$)=M THEN GOTO 330
270 PRINT
280 PRINT
290 PRINT "WRONG -- THE MISSING LETTER"
300 PRINT
310 PRINT "            IS -";CHR$ M;"-"
320 GOTO 350
330 PRINT
340 PRINT "*****CORRECT*****"
350 PRINT
360 PRINT "PLAY AGAIN Y/N"
370 INPUT Z$
380 IF Z$="Y" THEN RUN
390 STOP
```

Miner

BBC, ZX Spectrum, RML



You are the captain of an interplanetary mining ship, roaming the galaxy in search of precious minerals. Your ship's scanners inform you of an uninhabited planet nearby, which is rich in Andium ore. The glowing Andium ore is very rare and much sought after, so this discovery could mean a fortune for you and your crew. However, there is one terrifying drawback. An analysis of the planet reveals that its surface is highly explosive, making a landing virtually impossible. Your only hope is to try and land *exactly* on top of a deposit of ore, mine it quickly and take off again as fast as you can before disaster strikes. This is a tricky operation, calling for some expert ship-handling techniques. You also have to contend with the fact that Andium ore isn't easy to handle and so you have to go for the smallest deposits first.

Player keys

z – left
x – right
P – up

The Program

The first thing to do is to draw the explosive yellow surface of the uninhabited planet (line 40). (For the Spectrum you will need to tap in thirty-one spaces.)

BBC

```
   10 ON ERROR GOTO 370
   20 MODE7
   30 VDU23;8202;0;0;0;
   40 FOR R=11 TO 21:PRINTTAB(0,R);CHR$131;CHR$157;CHR$1
45:NEXT R
```

Spectrum

```
10 REM
20 BORDER 4:INK 2:PAPER 6
30 REM
40 FOR R=11 TO 21:PRINT AT R,0;"
      ":NEXT R
```

RML

```
10 REM
20 PUT 12
30 RANDOMIZE
40 FORR=11 TO 21:FOR C=1 TO 31:PLOT C*2,(21-R)*3,191:
NEXT C:NEXT R
```

The computer then sets up an array of thirty-two possible positions for the ore (line 50). It places ten lines, ranging in length from one to ten units, at random across the surface of the planet (line 70). On the BBC micro, we have to ask the computer to print 5 to get a line, because we are in Mode 7. On the ZX Spectrum we use graphic shift 5 (line 90).

BBC

```
 50 DIM L(32)
 60 FORR=1 TO 10
 70 LET X=RND(29)+2:IF L(X)>0 THEN GOTO 70
 80 LET Y=R:LET L(X)=R
 90 FOR Y=11 TO R+11:PRINTTAB(X,Y);"5":NEXT Y
100 NEXT R
```

Spectrum

```
 50 DIM L(32)
 60 FORR=1 TO 10
 70 LET X=RND(29)+2:IF L(X)>0 THEN GOTO 70
 80 LET Y=R:LET L(X)=R
 90 FOR Y=11 TO R+11:PRINT AT Y,X;"■":NEXT Y
100 NEXT R
```

RML

```
 50 DIM L(32)
 60 FOR R=1 TO 10
 70 LET X=INT(RND(1)*29)+3:IF L(X)>0 THEN GOTO 70
 80 LET Y=R:LET L(X)=R
 90 FOR Y=11 TO R+11:PLOT X*2,(21-Y)*3,255:NEXT Y
100 NEXT R
```

We then set the initial coordinates for your ship, tell the computer that you are aiming for the shortest line, and set the score to zero (line 110). Line 20 on the BBC sets the auto-repeat delay to a minimum, to help make the ship move as smoothly as possible. Finally, line 130 prints your score at the top of the screen.

BBC

```
110 LET MX=18:LET MY=0:LET R=1:LET SC=0
120 *FX11,1
```

Spectrum

```
110 LET MX=18:LET MY=0:LET R=1:LET SC=0
120 REM
```

RML

```
110 LET MX=18:LET MY=2:LET R=1:LET SC=0
120 REM
```

Now for the main program loop. The computer prints your mining ship (H) at coordinates MX,MY, and then checks to see whether the ship has landed on the correct line of ore (lines 150, 160). On the BBC and Spectrum, a strange, steam-engine sound is generated as the ship moves (line 180). The computer unplots your ship (line 200) and then checks to see whether you are pressing any directional keys (lines 210, 220, 230). Pressing z makes the ship move to the left; x makes it move to

the right and P makes it move up. Lines 240–260 prevent the ship going off the screen, while line 270 makes the ship descend automatically. And then the main loop begins again.

BBC

```
  130 PRINTTAB(MX,MY);"H"
  140 IF MY=11 AND L(MX)>0 THEN GOTO 290
  150 IF MY=11 AND L(MX)=0 THEN GOTO 360
  160 PRINTTAB(0,0);CHR$129;CHR$157;CHR$131;"YOU HAVE LA
NDED ON ";SC;" LINES"
  170 *FX15,1
  180 SOUND0,-8,4,1
  190 LET G$=INKEY$(10)
  200 PRINTTAB(MX,MY);" "
  210 IF G$="Z" THEN LET MX=MX-1
  220 IF G$="X" THEN LET MX=MX+1
  230 IF G$="P" THEN LET MY=MY-2
  240 IF MY<1 THEN LET MY=1
  250 IF MX<2 THEN LET MX=1
  260 IF MX>31 THEN LET MX=31
  270 LET MY=MY+1
  280 GOTO 130
```

Spectrum

```
  130 PRINT AT MY,MX;"H"
  140 IF MY=11 AND L(MX)>0 THEN GOTO 290
  150 IF MY=11 AND L(MX)=0 THEN GOTO 360
  160 PRINT AT 0,0;"YOU HAVE LANDED ON ";SC;" LINES"
  170 PAPER 7
  180 BEEP 0.1,12-MY
  190 LET G$=INKEY$
  200 PRINT AT MY,MX;" "
  210 IF G$="Z" THEN LET MX=MX-1
  220 IF G$="X" THEN LET MX=MX+1
  230 IF G$="P" THEN LET MY=MY-2
  240 IF MY<1 THEN LET MY=1
  250 IF MX<2 THEN LET MX=1
  260 IF MX>31 THEN LET MX=31
  270 LET MY=MY+1
  280 GOTO 130
```

RML

```
  130 PLOT MX*2,(21-MY)*3,"H"
  140 IF MY=11 AND L(MX)>0 THEN GOTO 290
  150 IF MY=11 AND L(MX)=0 THEN GOTO 360
```

```
160 PLOT 0,57,SC+245
170 REM
180 REM
190 LET G$=GET$(10)
200 PLOT MX*2,(21-MY)*3,32
210 IF G$="Z" THEN LET MX=MX-1
220 IF G$="X" THEN LET MX=MX+1
230 IF G$="P" THEN LET MY=MY-2
240 IF MY<1 THEN LET MY=1
250 IF MX<2 THEN LET MX=2
260 IF MX>31 THEN LET MX=31
270 LET MY=MY+1
280 GOTO 130
```

Checking to see whether your ship has landed on a line of ore is done at lines 150 and 160. If you have been successful, the computer jumps to a sub-loop at 290, where it checks the length of the line. If the line you landed on was the shortest, you win a point and jump back to the beginning of the main loop. If it was the wrong line, you lose.

BBC

```
290 IF L(MX)=R THEN LET SC=SC+1
300 IF L(MX)<>R THEN GOTO 360
310 IF R=10 THEN GOTO 390
320 FOR O=0 TO R:PRINTTAB(MX,O+11);" ":NEXT O
330 SOUND1,-15,200,1
340 LET MY=0:LET MX=18:LET R=R+1
350 GOTO 160
```

Spectrum

```
290 IF L(MX)=R THEN LET SC=SC+1
300 IF L(MX)<>R THEN GOTO 360
310 IF R=10 THEN GOTO 390
315 PAPER 6
320 FOR O=0 TO R:PRINT AT O+11,MX;" ":NEXT O
330 BEEP 1,40
340 LET MY=0:LET MX=18:LET R=R+1
350 GOTO 160
```

RML

```
290 IF L(MX)=R THEN LET SC=SC+1
300 IF L(MX)<>R THEN GOTO 360
310 IF R=10 THEN GOTO 390
320 FOR O=0 TO R:PLOT MX*2,(10-O)*3,191:NEXT O
330 REM
340 LET MY=0:LET MX=18:LET R=R+1
350 GOTO 160
```

If you lose, the joke's on you! This end routine for the BBC and Spectrum prints 'HA!' all over the screen, while the RML gets a spectacular character-set explosion.

BBC

```
360 FOR R=1 TO 100:PRINTTAB(RND(34)+1,RND(21));CHR$(12
8+RND(7));"HA!":NEXT R
370 *FX11,55
380 G$=GET$:IF G$<>"N" THEN RUN
385 END
```

Spectrum

```
360 FOR R=1 TO 100:PRINT AT INT(RND*21),INT(RND*26);"H
A!":NEXT R
370 REM
380 INPUT G$:RUN
```

RML

```
360 FOR R=1 TO 60:PLOT MX*2,(21-MY)*3,R+195:NEXT R
370 REM
380 G$=GET$():IF G$<>"N" THEN RUN
385 END
```

And if you win, the computer has a better message for you: this little loop prints 'BRILLIANT!' all over the screen.

BBC

```
  390 FOR R=1 TO 60:PRINTTAB(RND(26)+2,RND(21));CHR$(128
+RND(7));"BRILLIANT!":NEXT R
  400 GOTO 370
```

Spectrum

```
  390 FOR R=1 TO 60:PRINT AT INT(RND*21),INT(RND*24);"BR
ILLIANT!":NEXT R
  400 GOTO 370
```

RML

```
390 PLOT MX*2-8,(22-MY)*3,"BRILLIANT"
400 GOTO 370
```

Suggestions

Try getting the player to input a level of difficulty early on in the program by creating a variable for the descent of the ship in line 190.

Full Program: BBC

```
   10 ON ERROR GOTO 370
   20 MODE7
   30 VDU23;8202;0;0;0;
   40 FOR R=11 TO 21:PRINTTAB(0,R);CHR$131;CHR$157;CHR$1
45:NEXT R
   50 DIM L(32)
   60 FORR=1 TO 10
   70 LET X=RND(29)+2:IF L(X)>0 THEN GOTO 70
   80 LET Y=R:LET L(X)=R
   90 FOR Y=11 TO R+11:PRINTTAB(X,Y);"5":NEXT Y
  100 NEXT R
  110 LET MX=18:LET MY=0:LET R=1:LET SC=0
  120 *FX11,1
  130 PRINTTAB(MX,MY);"H"
  140 IF MY=11 AND L(MX)>0 THEN GOTO 290
  150 IF MY=11 AND L(MX)=0 THEN GOTO 360
  160 PRINTTAB(0,0);CHR$129;CHR$157;CHR$131;"YOU HAVE LA
NDED ON ";SC;" LINES"
  170 *FX15,1
  180 SOUND0,-8,4,1
  190 LET G$=INKEY$(10)
  200 PRINTTAB(MX,MY);" "
  210 IF G$="Z" THEN LET MX=MX-1
  220 IF G$="X" THEN LET MX=MX+1
  230 IF G$="P" THEN LET MY=MY-2
  240 IF MY<1 THEN LET MY=1
  250 IF MX<2 THEN LET MX=1
  260 IF MX>31 THEN LET MX=31
  270 LET MY=MY+1
  280 GOTO 130
  290 IF L(MX)=R THEN LET SC=SC+1
  300 IF L(MX)<>R THEN GOTO 360
  310 IF R=10 THEN GOTO 390
  320 FOR O=0 TO R:PRINTTAB(MX,O+11);" ":NEXT O
  330 SOUND1,-15,200,1
  340 LET MY=0:LET MX=18:LET R=R+1
  350 GOTO 160
  360 FOR R=1 TO 100:PRINTTAB(RND(34)+1,RND(21));CHR$(12
8+RND(7));"HA!":NEXT R
  370 *FX11,55
  380 G$=GET$:IF G$<>"N" THEN RUN
  385 END
  390 FOR R=1 TO 60:PRINTTAB(RND(26)+2,RND(21));CHR$(128
+RND(7));"BRILLIANT!":NEXT R
  400 GOTO 370
```

Full Program: ZX Spectrum

```
  10 REM
  20 BORDER 4:INK 2:PAPER 6
  30 REM
  40 FOR R=11 TO 21:PRINT AT R,0;"
       ":NEXT R
  50 DIM L(32)
  60 FORR=1 TO 10
  70 LET X=RND(29)+2:IF L(X)>0 THEN GOTO 70
  80 LET Y=R:LET L(X)=R
  90 FOR Y=11 TO R+11:PRINT AT Y,X;"■":NEXT Y
 100 NEXT R
 110 LET MX=18:LET MY=0:LET R=1:LET SC=0
 120 REM
 130 PRINT AT MY,MX;"H"
 140 IF MY=11 AND L(MX)>0 THEN GOTO 290
 150 IF MY=11 AND L(MX)=0 THEN GOTO 360
 160 PRINT AT 0,0;"YOU HAVE LANDED ON ";SC;" LINES"
 170 PAPER 7
 180 BEEP 0.1,12-MY
 190 LET G$=INKEY$
 200 PRINT AT MY,MX;" "
 210 IF G$="Z" THEN LET MX=MX-1
 220 IF G$="X" THEN LET MX=MX+1
 230 IF G$="P" THEN LET MY=MY-2
 240 IF MY<1 THEN LET MY=1
 250 IF MX<2 THEN LET MX=1
 260 IF MX>31 THEN LET MX=31
 270 LET MY=MY+1
 280 GOTO 130
 290 IF L(MX)=R THEN LET SC=SC+1
 300 IF L(MX)<>R THEN GOTO 360
 310 IF R=10 THEN GOTO 390
 315 PAPER 6
 320 FOR 0=0 TO R:PRINT AT 0+11,MX;" ":NEXT 0
 330 BEEP 1,40
 340 LET MY=0:LET MX=18:LET R=R+1
 350 GOTO 160
 360 FOR R=1 TO 100:PRINT AT INT(RND*21),INT(RND*26);"H
A!":NEXT R
 370 REM
 380 INPUT G$:RUN
 390 FOR R=1 TO 60:PRINT AT INT(RND*21),INT(RND*24);"BR
ILLIANT!":NEXT R
 400 GOTO 370
```

Full Program: RML

```
   10 REM
   20 PUT 12
   30 RANDOMIZE
   40 FORR=11 TO 21:FOR C=1 TO 31:PLOT C*2,(21-R)*3,191:
NEXT C:NEXT R
   50 DIM L(32)
   60 FOR R=1 TO 10
   70 LET X=INT(RND(1)*29)+3:IF L(X)>0 THEN GOTO 70
   80 LET Y=R:LET L(X)=R
   90 FOR Y=11 TO R+11:PLOT X*2,(21-Y)*3,255:NEXT Y
  100 NEXT R
  110 LET MX=18:LET MY=2:LET R=1:LET SC=0
  120 REM
  130 PLOT MX*2,(21-MY)*3,"H"
  140 IF MY=11 AND L(MX)>0 THEN GOTO 290
  150 IF MY=11 AND L(MX)=0 THEN GOTO 360
  160 PLOT 0,57,SC+245
  170 REM
  180 REM
  190 LET G$=GET$(10)
  200 PLOT MX*2,(21-MY)*3,32
  210 IF G$="Z" THEN LET MX=MX-1
  220 IF G$="X" THEN LET MX=MX+1
  230 IF G$="P" THEN LET MY=MY-2
  240 IF MY<1 THEN LET MY=1
  250 IF MX<2 THEN LET MX=2
  260 IF MX>31 THEN LET MX=31
  270 LET MY=MY+1
  280 GOTO 130
  290 IF L(MX)=R THEN LET SC=SC+1
  300 IF L(MX)<>R THEN GOTO 360
  310 IF R=10 THEN GOTO 390
  320 FOR O=0 TO R:PLOT MX*2,(10-O)*3,191:NEXT O
  330 REM
  340 LET MY=0:LET MX=18:LET R=R+1
  350 GOTO 160
  360 FOR R=1 TO 60:PLOT MX*2,(21-MY)*3,R+195:NEXT R
  370 REM
  380 G$=GET$():IF G$<>"N" THEN RUN
  385 END
  390 PLOT MX*2-8,(22-MY)*3,"BRILLIANT"
  400 GOTO 370
```

Shooting in the Dark

ZX81

How expert a marksman are you? Can you hit a moving target accurately? This game is designed to find out just how good you are. The computer marks out x and y axes for you to work from. Your target appears and moves. You take aim but, much to your surprise, your target suddenly disappears! It hasn't stopped and your task is to guess exactly where it has got to. Key in the y-coordinate for your guess and the computer will tell you whether you are right. Just see how long it takes you to hit a bull's-eye!

The Program

We will start by giving the computer some data to work from. X and Y are the target's coordinates, E is its x-direction, G is the number of goes you take, and H is a flag which tells the computer if the target should be visible or not.

```
 10 RAND
 20 SLOW
 30 LET X=3
 40 LET Y=23
 50 LET E=1
 60 LET G=1
 70 LET H=0
```

Then we draw the x and y axes, which will help you with your coordinate guesses. Lines 80–100 draw the x-axis and lines 110–140 label it. The y-axis is drawn and labelled in lines 150–200. The last thing we have to do before the main loop is to give the target's y-direction a value (line 210).

```
 80 FOR R=2 TO 62
 90 PLOT R,3
100 NEXT R
110 FOR R=2 TO 62 STEP 10
120 PLOT R,2
130 PRINT AT 21,R/2-1;R-2
140 NEXT R
150 FOR R=3 TO 43
160 PLOT 2,R
170 NEXT R
```

```
180 FOR R=3 TO 43 STEP 10
190 PLOT 1,R
200 NEXT R
210 LET D=RND*4/2-1
```

The main part of this program is a double loop from line 220 to line 400. The inner loop moves the target on the screen for 29 spaces, checking whether it hits either side or the top or bottom of the screen at lines 260–270. Once it has completed the first 29 moves, H becomes 1 and the program makes a further 29 loops, missing out the plot line at 290. The outer loop takes over to ask you for your guess for the target's new y-coordinate (line 350), works out whether you are right or wrong (lines 370–380), and adds one to the number of guesses you have had (line 390) before returning to the beginning of the main loop again.

```
220 FOR R=1 TO 29
230 UNPLOT X,Y
240 LET Y=Y+D
250 LET X=X+E
260 IF X<4 OR X>61 THEN LET E=-E
270 IF Y<5 OR Y>42 THEN LET D=-D
280 IF H=1 THEN GOTO 300
290 PLOT X,Y
300 NEXT R
310 IF H=1 THEN GOTO 340
320 LET H=1
330 GOTO 220
340 LET H=0
350 PRINT AT 19,2;"Y CO.ORD?                "
360 INPUT A
370 IF A=INT(Y)-3 THEN GOTO 410
380 PRINT AT 19,2;"--WRONG-- IT WAS ";INT(Y)-3
390 LET G=G+1
400 GOTO 220
```

If you hit the target fair and square, the computer will tell you how many goes you took.

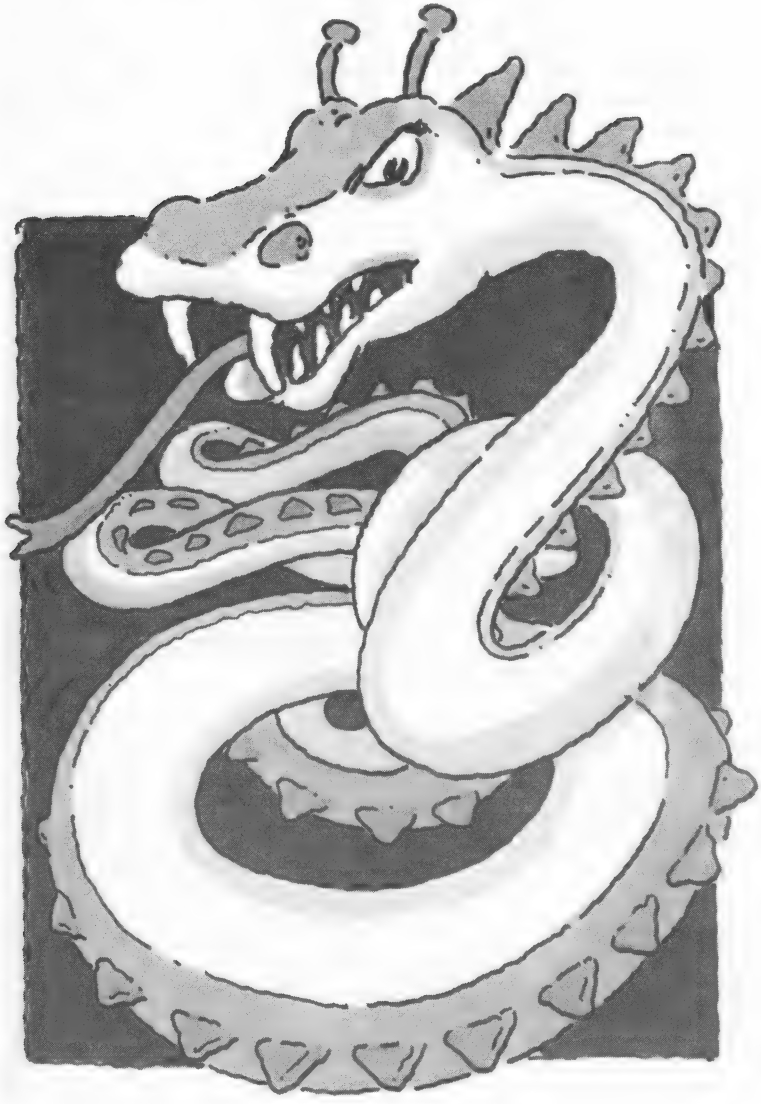
```
410 PRINT AT 19,2;"*CORRECT* YOU TOOK ";G;" GOES"
420 STOP
```

Full Program

```
 10 RAND
 20 SLOW
 30 LET X=3
 40 LET Y=23
 50 LET E=1
 60 LET G=1
 70 LET H=0
 80 FOR R=2 TO 62
 90 PLOT R,3
100 NEXT R
110 FOR R=2 TO 62 STEP 10
120 PLOT R,2
130 PRINT AT 21,R/2-1;R-2
140 NEXT R
150 FOR R=3 TO 43
160 PLOT 2,R
170 NEXT R
180 FOR R=3 TO 43 STEP 10
190 PLOT 1,R
200 NEXT R
210 LET D=RND*4/2-1
220 FOR R=1 TO 29
230 UNPLOT X,Y
240 LET Y=Y+D
250 LET X=X+E
260 IF X<4 OR X>61 THEN LET E=-E
270 IF Y<5 OR Y>42 THEN LET D=-D
280 IF H=1 THEN GOTO 300
290 PLOT X,Y
300 NEXT R
310 IF H=1 THEN GOTO 340
320 LET H=1
330 GOTO 220
340 LET H=0
350 PRINT AT 19,2;"Y CO.ORD?              "
360 INPUT A
370 IF A=INT(Y)-3 THEN GOTO 410
380 PRINT AT 19,2;"--WRONG-- IT WAS ";INT(Y)-3
390 LET G=G+1
400 GOTO 220
410 PRINT AT 19,2;"*CORRECT* YOU TOOK ";G;" GOES"
420 STOP
```

Snake Runner

BBC, ZX Spectrum, RML



You have been trapped by a strange, snake-like creature which runs round and round you. It can't reach you, but you can't leave either. Your only hope is to try to kill the creature outright, but you know that its only vulnerable spot is in its tail. You are armed with a gun and forty bullets and have to try to shoot off each of the tail segments in turn, starting from the end. This is far from easy and you must use your bullets sparingly. If you are careful you can control your bullet's direction, but you have only a very limited space to work in.

Player keys

Z – left
X – right
K – up
M – down

The Program

Some basic instructions to start with: for BBC users we turn off the auto-repeat (line 10), go to the teletext mode (line 20) and turn off the cursor (line 30); for Spectrum users, all we have to do is to define the colours we are using. In this program the border is blue and the game is white on black (line 20); for the RML micros we go to graphics (line 20).

BBC

```
10 *FX11,0
20 MODE7
30 VDU23;8202;0;0;0;
```

Spectrum

```
10 REM
20 BORDER 1:PAPER 0:INK 7:CLS
30 REM
```

RML

```
10 REM
20 GRAPH
30 REM
```

The first major task is to set up the rectangular course for the snake (lines 50–90). Line 40 gives the computer data to work from and sets up the dimensions of the course, allowing it to overlap down one side so that the snake will run smoothly. MX and MY are the coordinates of the player's character, A and B are the direction of the bullet, L counts the number of segments in the snake, BX and BY are the coordinates of the bullet, S is the number of shots used.

BBC, Spectrum and RML

```
   40 DIM X(102):DIM Y(102):LET C=1:LET MX=15:LET MY=11:
LET A=0:LET B=0:LET L=0:LET BX=15:LET BY=11:LET S=0
   50 FORR=29 TO 2 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=C
+1:NEXT R
   60 FORR=3 TO 19:LET X(C)=2:LET Y(C)=R:LET C=C+1:NEXT
R
   70 FORR=2 TO 29:LET X(C)=R:LET Y(C)=20:LET C=C+1:NEXT
 R
   80 FORR=19 TO 3 STEP -1:LET X(C)=29:LET Y(C)=R:LET C=
C+1:NEXT R
   90 FORR=29 TO 18 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=
C+1:NEXT R
```

The last thing we do before going into the main loop is to get your score printed up, together with the number of shots remaining to you.

BBC

```
  100 PRINT TAB(0,0);CHR$130;10-L;" SEGMENTS";TAB(20);CH
R$131;40-S;" SHOTS"
```

Spectrum

```
  100 PRINT AT 0,0;10-L;" SEGMENTS";TAB(20);40-S;" SHOTS
"
```

RML

```
  100 PUT 12:PRINT TAB(0);10-L;" SEGMENTS ";TAB(20);40-S
;" SHOTS "
```

The main loop runs from line 110 to 230 and round again. Line 110 prints the player's character and line 120 jumps out to a small subroutine at line 330 to move the snake and then jumps back again. Lines 140–180 check which keys are being pressed and give values for A and B, which will be used to move the bullet, and line 190 jumps to the shooting routine at line 240. The computer checks whether you have hit the end segment of the snake at line 200 and, if you have and it was the last segment left, tells you just how brilliant you are before sending you to the end routine at line 350. Finally, line 220 gives a new value for R, which will be used to move the snake round again.

BBC

```
  110 PRINT TAB(MX,MY);"#"
  120 GOSUB 330
  130 SOUND4,-5,10,1
  140 LET G$=INKEY$((10-L)/2)
  150 IF G$="Z" THEN LET A=-1
  160 IF G$="X" THEN LET A=1
  170 IF G$="M" THEN LET B=1
  180 IF G$="K" THEN LET B=-1
  190 IF A<>0 OR B<>0 THEN GOSUB 240
  200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:VDU7
:PRINT TAB(0,0);CHR$130;10-L;" SEGMENTS "
  210 IF L>9 THEN PRINT TAB(11,MY-2);"BRILLIANT!":GOTO 3
50
  220 LET R=R+1:IF R>102 THEN LET R=13
  230 GOTO 110
```

Spectrum

```
  110 PRINT AT MY,MX;"#"
  120 GOSUB 330
  130 BEEP 0.01,5
  140 LET G$=INKEY$
  150 IF G$="Z" THEN LET A=-1
  160 IF G$="X" THEN LET A=1
  170 IF G$="M" THEN LET B=1
  180 IF G$="K" THEN LET B=-1
  190 IF A<>0 OR B<>0 THEN GOSUB 240
  200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:BEEP
0.1,30:PRINT AT 0,0;10-L;" SEGMENTS "
  210 IF L>9 THEN PRINT AT MY-2,11;"BRILLIANT!":GOTO 350
  220 LET R=R+1:IF R>102 THEN LET R=13
  230 GOTO 110
```

RML

```
  110 PLOT MX*2,(21-MY)*3,"*"
  120 GOSUB 330
  130 REM
  140 LET G$=GET$((19-L)/2)
  150 IF G$="Z" THEN LET A=-1
  160 IF G$="X" THEN LET A=1
  170 IF G$="M" THEN LET B=1
  180 IF G$="K" THEN LET B=-1
  190 IF A<>0 OR B<>0 THEN GOSUB 240
  200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:PRIN
T TAB(0);10-L;" SEGMENTS ";
  210 IF L>9 THEN PLOT 22,(19-MY)*3,"BRILLIANT!":GOTO 35
0
  220 LET R=R+1:IF R>102 THEN LET R=13
  230 GOTO 110
```

Line 190 sends the computer to this shooting subroutine at lines 240–280, which erases, moves and reprints the bullet (a +), using the values for A and B from the main loop. Lines 260 and 270 check whether the bullet has left the screen and, if so, jump to line 300 for another bullet.

BBC

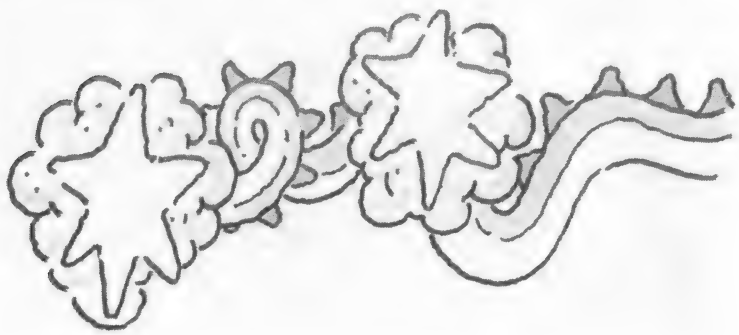
```
240 PRINT TAB(BX,BY);" "
250 LET BX=BX+A:LET BY=BY+B
260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
280 PRINT TAB(BX,BY);"+"
290 RETURN
```

Spectrum

```
240 PRINT AT BY,BX;" "
250 LET BX=BX+A:LET BY=BY+B
260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
280 PRINT AT BY,BX;"+"
290 RETURN
```

RML

```
240 PLOT BX*2,(21-BY)*3," "
250 LET BX=BX+A:LET BY=BY+B
260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
280 PLOT BX*2,(21-BY)*3,"+"
290 RETURN
```



When one of the bullets leaves the screen (lines 260, 270), this next subroutine starts the next bullet from the player's character and adds one to s, which deducts one from the bullets left. If you have used all forty shots, you have come to the end of the game and the computer jumps to the end routine at line 350. Finally, line 330 moves the snake runner along its course.

BBC

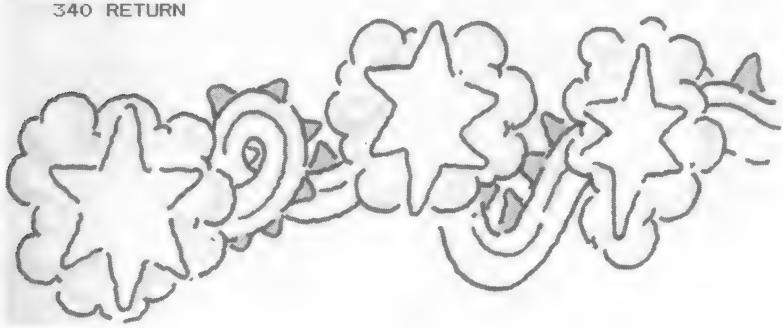
```
  300 LET BX=MX:LET BY=MY:LET S=S+1
  310 PRINT TAB(20,0);CHR$131;40-S;" SHOTS "
  320 IF S>39 THEN GOTO 350
  330 PRINT TAB(X(R),Y(R));"O":PRINT TAB(X(R-10+L),Y(R-1
0+L));" "
  340 RETURN
```

Spectrum

```
  300 LET BX=MX:LET BY=MY:LET S=S+1
  310 PRINT AT 0,20;40-S;" SHOTS "
  320 IF S>39 THEN GOTO 350
  330 PRINT AT Y(R),X(R);"O":PRINT AT Y(R-10+L),X(R-10+L
);" "
  340 RETURN
```

RML

```
  300 LET BX=MX:LET BY=MY:LET S=S+1
  310 PRINT TAB(20);40-S;" SHOTS "
  320 IF S>39 THEN GOTO 350
  330 PLOT X(R)*2,(21-Y(R))*3,"O":PLOT X(R-10+L)*2,(21-Y
(R-10+L))*3," "
  340 RETURN
```



As usual, the last routine in the program is the one where you, the player, are offered another game.

BBC

```
350 PRINT TAB(8,18);"PLAY AGAIN Y/N";
360 INPUT A$
370 IF A$="Y" OR A$="YES" THEN RUN
380 *FX11,55
390 STOP
```

Spectrum

```
350 PRINT AT 18,8;"PLAY AGAIN Y/N?";
360 INPUT A$
370 IF A$="Y" OR A$="YES" THEN RUN
380 REM
390 STOP
```

RML

```
350 PLOT 19,9,"PLAY AGAIN Y/N?"
360 INPUT A$
370 IF A$="Y" OR A$="YES" THEN RUN
380 TEXT
390 END
```

Suggestions

To make the game more difficult, you could allow yourself fewer shots (line 320). Giving yourself only twenty shots would really test your mettle!

If you feel really ambitious, try changing the shape of the course by adding more changes of direction between lines 40–90. It would be simplest to stick to the same number of steps used in this program, but, if you want to use a few more, remember to change DIM X and DIM Y, and line 220.

Full Program: BBC

```
  10 *FX11,0
  20 MODE7
  30 VDU23;8202;0;0;0;
  40 DIM X(102):DIM Y(102):LET C=1:LET MX=15:LET MY=11:
LET A=0:LET B=0:LET L=0:LET BX=15:LET BY=11:LET S=0
  50 FORR=29 TO 2 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=C
+1:NEXT R
  60 FORR=3 TO 19:LET X(C)=2:LET Y(C)=R:LET C=C+1:NEXT
R
  70 FORR=2 TO 29:LET X(C)=R:LET Y(C)=20:LET C=C+1:NEXT
 R
  80 FORR=19 TO 3 STEP -1:LET X(C)=29:LET Y(C)=R:LET C=
C+1:NEXT R
  90 FORR=29 TO 18 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=
C+1:NEXT R
 100 PRINT TAB(0,0);CHR$130;10-L;" SEGMENTS";TAB(20);CH
R$131;40-S;" SHOTS"
 110 PRINT TAB(MX,MY);"#"
 120 GOSUB 330
 130 SOUND4,-5,10,1
 140 LET G$=INKEY$((10-L)/2)
 150 IF G$="Z" THEN LET A=-1
 160 IF G$="X" THEN LET A=1
 170 IF G$="M" THEN LET B=1
 180 IF G$="K" THEN LET B=-1
 190 IF A<>0 OR B<>0 THEN GOSUB 240
 200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:VDU7
:PRINT TAB(0,0);CHR$130;10-L;" SEGMENTS "
 210 IF L>9 THEN PRINT TAB(11,MY-2);"BRILLIANT!":GOTO 3
50
 220 LET R=R+1:IF R>102 THEN LET R=13
 230 GOTO 110
 240 PRINT TAB(BX,BY);" "
 250 LET BX=BX+A:LET BY=BY+B
 260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
 270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
 280 PRINT TAB(BX,BY);"+"
 290 RETURN
 300 LET BX=MX:LET BY=MY:LET S=S+1
 310 PRINT TAB(20,0);CHR$131;40-S;" SHOTS "
 320 IF S>39 THEN GOTO 350
 330 PRINT TAB(X(R),Y(R));"O":PRINT TAB(X(R-10+L),Y(R-1
0+L));" "
 340 RETURN
 350 PRINT TAB(8,18);"PLAY AGAIN Y/N";
 360 INPUT A$
 370 IF A$="Y" OR A$="YES" THEN RUN
 380 *FX11,55
 390 STOP
```

Full Program: Spectrum

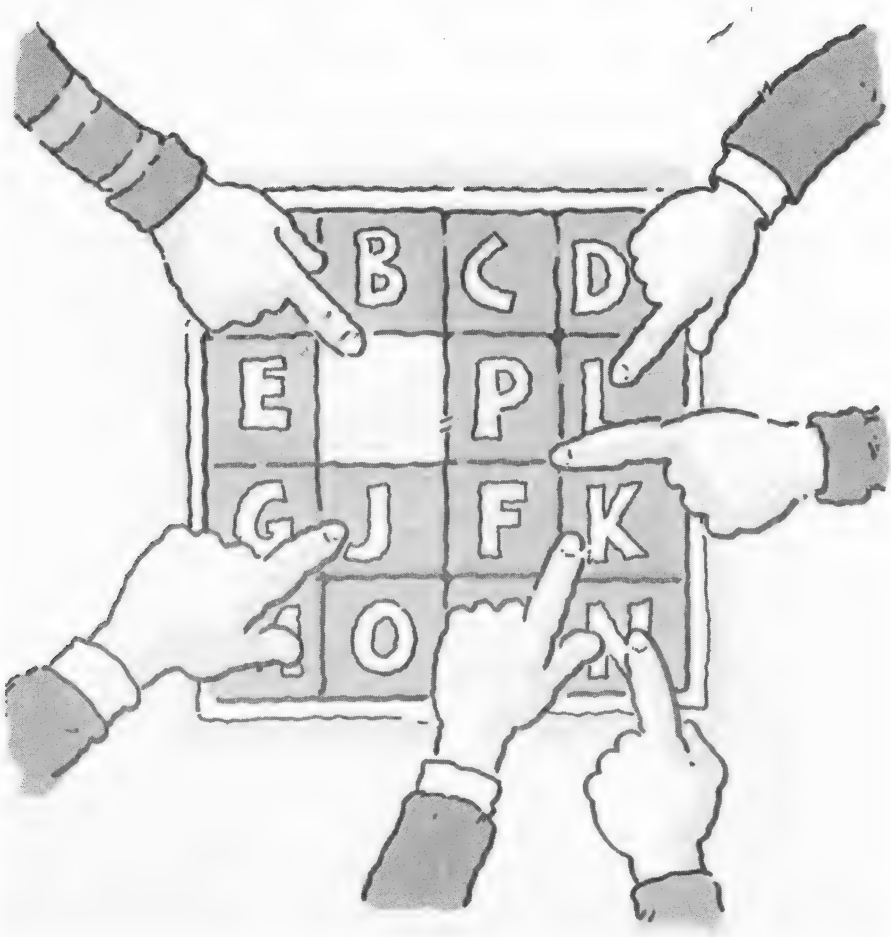
```
   10 REM
   20 BORDER 1:PAPER 0:INK 7:CLS
   30 REM
   40 DIM X(102):DIM Y(102):LET C=1:LET MX=15:LET MY=11:
LET A=0:LET B=0:LET L=0:LET BX=15:LET BY=11:LET S=0
   50 FORR=29 TO 2 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=C
+1:NEXT R
   60 FORR=3 TO 19:LET X(C)=2:LET Y(C)=R:LET C=C+1:NEXT
R
   70 FORR=2 TO 29:LET X(C)=R:LET Y(C)=20:LET C=C+1:NEXT
 R
   80 FORR=19 TO 3 STEP -1:LET X(C)=29:LET Y(C)=R:LET C=
C+1:NEXT R
   90 FORR=29 TO 18 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=
C+1:NEXT R
  100 PRINT AT 0,0;10-L;" SEGMENTS";TAB(20);40-S;" SHOTS
"
  110 PRINT AT MY,MX;"#"
  120 GOSUB 330
  130 BEEP 0.01,5
  140 LET G$=INKEY$
  150 IF G$="Z" THEN LET A=-1
  160 IF G$="X" THEN LET A=1
  170 IF G$="M" THEN LET B=1
  180 IF G$="K" THEN LET B=-1
  190 IF A<>0 OR B<>0 THEN GOSUB 240
  200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:BEEP
 0.1,30:PRINT AT 0,0;10-L;" SEGMENTS "
  210 IF L>9 THEN PRINT AT MY-2,11;"BRILLIANT!":GOTO 350
  220 LET R=R+1:IF R>102 THEN LET R=13
  230 GOTO 110
  240 PRINT AT BY,BX;" "
  250 LET BX=BX+A:LET BY=BY+B
  260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
  270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
  280 PRINT AT BY,BX;"+"
  290 RETURN
  300 LET BX=MX:LET BY=MY:LET S=S+1
  310 PRINT AT 0,20;40-S;" SHOTS "
  320 IF S>39 THEN GOTO 350
  330 PRINT AT Y(R),X(R);"O":PRINT AT Y(R-10+L),X(R-10+L
);" "
  340 RETURN
  350 PRINT AT 18,8;"PLAY AGAIN Y/N?";
  360 INPUT A$
  370 IF A$="Y" OR A$="YES" THEN RUN
  380 REM
  390 STOP
```

Full Program: RML

```
   10 REM
   20 GRAPH
   30 REM
   40 DIM X(102):DIM Y(102):LET C=1:LET MX=15:LET MY=11:
LET A=0:LET B=0:LET L=0:LET BX=15:LET BY=11:LET S=0
   50 FORR=29 TO 2 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=C
+1:NEXT R
   60 FORR=3 TO 19:LET X(C)=2:LET Y(C)=R:LET C=C+1:NEXT
R
   70 FORR=2 TO 29:LET X(C)=R:LET Y(C)=20:LET C=C+1:NEXT
 R
   80 FORR=19 TO 3 STEP -1:LET X(C)=29:LET Y(C)=R:LET C=
C+1:NEXT R
   90 FORR=29 TO 18 STEP -1:LET X(C)=R:LET Y(C)=2:LET C=
C+1:NEXT R
  100 PUT 12:PRINT TAB(0);10-L;" SEGMENTS ";TAB(20);40-S
;" SHOTS "
  110 PLOT MX*2,(21-MY)*3,"*"
  120 GOSUB 330
  130 REM
  140 LET G$=GET$((19-L)/2)
  150 IF G$="Z" THEN LET A=-1
  160 IF G$="X" THEN LET A=1
  170 IF G$="M" THEN LET B=1
  180 IF G$="K" THEN LET B=-1
  190 IF A<>0 OR B<>0 THEN GOSUB 240
  200 IF BX=X(R-9+L) AND BY=Y(R-9+L) THEN LET L=L+1:PRIN
T TAB(0);10-L;" SEGMENTS ";
  210 IF L>9 THEN PLOT 22,(19-MY)*3,"BRILLIANT!":GOTO 35
0
  220 LET R=R+1:IF R>102 THEN LET R=13
  230 GOTO 110
  240 PLOT BX*2,(21-BY)*3," "
  250 LET BX=BX+A:LET BY=BY+B
  260 IF BX<2 OR BX>29 THEN LET A=0:GOTO 300
  270 IF BY<2 OR BY>20 THEN LET B=0:GOTO 300
  280 PLOT BX*2,(21-BY)*3,"+"
  290 RETURN
  300 LET BX=MX:LET BY=MY:LET S=S+1
  310 PRINT TAB(20);40-S;" SHOTS "
  320 IF S>39 THEN GOTO 350
  330 PLOT X(R)*2,(21-Y(R))*3,"O":PLOT X(R-10+L)*2,(21-Y
(R-10+L))*3," "
  340 RETURN
  350 PLOT 19,9,"PLAY AGAIN Y/N?"
  360 INPUT A$
  370 IF A$="Y" OR A$="YES" THEN RUN
  380 TEXT
  390 END
```

Magic Square

BBC, ZX Spectrum, RML, ZX81



This is a mind-boggling game for two to play. The computer sets up a magic square of 4×4 letters in alphabetical order, with one gap. One player closes his or her eyes, while the other carefully muddles up all the letters. The first player then has to work out the quickest and simplest way of getting the square back into alphabetical order. The computer counts the number of moves taken to muddle and sort out the letters, and it is the difference between the two that really counts.

Player keys

Z – right
X – left
K – up
M – down

The Program

First of all, we have to give the computer some basic information, including the dimensions of both the player's square – S(5,5) – and the matrix the computer will use to check it with – B(5,5) (line 30, *20–80 ZX81*). P is the value of the characters in the square, M is the number of moves you make, K and L are coordinates used to swap pieces over, and N is a flag to check that the first move has taken place. Then we get some simple instructions printed up.

BBC

```
  10 MODE7
  20 VDU23;8202;0;0;0;
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PRINT TAB(6,6);"PUSH ANY KEY TO START"
  50 PRINT TAB(3,10);"SPACE BAR RESETS MOVES TO 0"
  60 PRINT TAB(12,15);"Q = QUIT"
  70 LET G$=INKEY$(3):IF G$="" THEN GOTO 70
  80 CLS
```

Spectrum

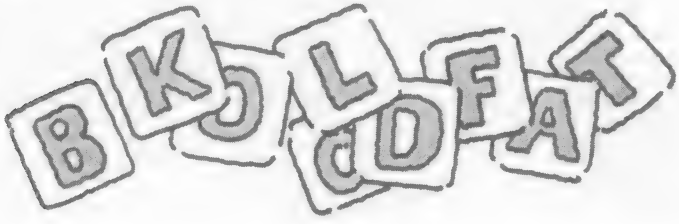
```
  10 BORDER 2:PAPER 0:INK 7:CLS
  20 REM
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PRINT AT 6,6;"PUSH ANY KEY TO START"
  50 PRINT AT 10,3;"SPACE BAR RESETS MOVES TO 0"
  60 PRINT AT 15,12;"Q = QUIT"
  70 LET G$=INKEY$:IF G$="" THEN GOTO 70
  80 CLS
```

RML

```
  10 GRAPH
  20 PUT 12
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PLOT 12,45,"PUSH ANY KEY TO START"
  50 PLOT 6,43,"SPACE BAR RESETS MOVES TO 0"
  60 PLOT 24,18,"Q = QUIT"
  70 LET G$=GET$()
  80 GRAPH
```

ZX81

```
  10 SLOW
  20 LET P=38
  30 DIM B(5,5)
  40 DIM S(5,5)
  50 LET M=0
  60 LET K=0
  70 LET L=0
  80 LET N=1
  90 PRINT AT 6,6;"PRESS ANY KEY TO START"
 100 PRINT AT 10,3;"THE P KEY RESETS MOVES TO 0"
 110 PRINT AT 15,12;"Q = QUIT"
 120 LET G$=INKEY$
 130 IF G$="" THEN GOTO 120
 140 CLS
```



The computer's next task is to set up the two matrices for the square – one for you, the player, and one for the computer to check against yours. Line 150 (*220–250 ZX81*) gives you a blank square in the bottom right-hand corner.

BBC, Spectrum, RML

```
 90 FOR R=1 TO 4
100 FOR S=1 TO 4
110 LET S(S,R)=P
120 LET B(S,R)=P
130 LET P=P+1
140 NEXT S:NEXT R
150 LET S(4,4)=32:LET B(4,4)=32:LET X=4:LET Y=4
```

ZX81

```
150 FAST
160 FOR R=1 TO 4
170 FOR S=1 TO 4
180 LET S(S,R)=P
190 LET B(S,R)=P
200 LET P=P+1
210 NEXT S
215 NEXT R
220 LET S(4,4)=0
230 LET B(4,4)=0
240 LET X=4
250 LET Y=4
```

Then comes the main program loop, which starts with the computer printing the player's matrix (line 160, *260–300 ZX81*). The two matrices are then checked against each other, piece by piece, and each correct letter adds one to Q (lines 180–200, *320–380 ZX81*). If all the pieces are in the right places, the program jumps out of the main loop to tell you that you have won (line 210, *390 ZX81*). Lines 230–290 (410–520 ZX81) check which keys you are pressing and move a piece accordingly (line 310, *530–590 ZX81*), before returning to the beginning of the main loop again.

BBC

```
  160 FORR=1 TO 4:FOR S=1 TO 4:PRINT TAB(S*6-1,R*4);CHR$
(129+(S(S,R) MOD 7));CHR$(S(S,R)):NEXT S:NEXT R
  170 PRINT TAB(11,1);M;" MOVES  "
  180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
 THEN LET Q=Q+1
  190 NEXT S:NEXT R
  200 PRINT TAB(11,20);Q;" CORRECT "
  210 IF Q=16 AND N=0 THEN GOTO 330
  220 *FX15,1
  230 LET G$=INKEY$(3):IF G$="" THEN GOTO 230
  240 IF G$="Z" THEN LET K=X+1:LET L=Y
  250 IF G$="X" THEN LET K=X-1:LET L=Y
  260 IF G$="K" THEN LET K=X:LET L=Y+1
  270 IF G$="M" THEN LET K=X:LET L=Y-1
  280 IF G$="Q" THEN GOTO 340
  290 IF G$=" " THEN LET M=-1
  300 IF S(K,L)=0 THEN GOTO 230
  310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
  320 GOTO 160
```

Spectrum

```
  160 FORR=1 TO 4:FOR S=1 TO 4:INK R+S:PRINT AT R*4,S*6-
1;CHR$(S(S,R)):NEXT S:NEXT R
  170 PRINT AT 1,11;M;" MOVES  "
  180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
 THEN LET Q=Q+1
  190 NEXT S:NEXT R
  200 PRINT AT 20,11;Q;" CORRECT "
  210 IF Q=16 AND N=0 THEN GOTO 330
  220 REM
  230 LET G$=INKEY$:IF G$="" THEN GOTO 230
  240 IF G$="Z" THEN LET K=X+1:LET L=Y
  250 IF G$="X" THEN LET K=X-1:LET L=Y
  260 IF G$="K" THEN LET K=X:LET L=Y+1
  270 IF G$="M" THEN LET K=X:LET L=Y-1
  280 IF G$="Q" THEN GOTO 340
  290 IF G$=" " THEN LET M=-1
  300 IF S(K,L)=0 THEN GOTO 230
  310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
  320 GOTO 160
```

RML

```
  160 FORR=1 TO 4:FOR S=1 TO 4:PLOT S*12-2,(21-R*4)*3,S(
S,R):NEXT S:NEXT R
  170 PUT 12:PRINT TAB(11);M;" MOVES  "
  180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
 THEN LET Q=Q+1
  190 NEXT S:NEXT R
  200 PRINT TAB(10);Q;" CORRECT "
  210 IF Q=16 AND N=0 THEN GOTO 330
  220 REM
  230 LET G$=GET$()
  240 IF G$="Z" THEN LET K=X+1:LET L=Y
  250 IF G$="X" THEN LET K=X-1:LET L=Y
  260 IF G$="K" THEN LET K=X:LET L=Y+1
  270 IF G$="M" THEN LET K=X:LET L=Y-1
  280 IF G$="Q" THEN GOTO 340
  290 IF G$=" " THEN LET M=-1
  300 IF S(K,L)=0 THEN GOTO 230
  310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
  320 GOTO 160
```

ZX81

```
  260 FOR R=1 TO 4
  270 FOR S=1 TO 4
  280 PRINT AT R*2,S*6;CHR$ S(S,R)
  290 NEXT S
  300 NEXT R
  310 PRINT AT 1,11;M;" MOVES  "
  320 LET Q=0
  330 FOR R=1 TO 4
  340 FOR S=1 TO 4
  350 IF S(S,R)=B(S,R) THEN LET Q=Q+1
  360 NEXT S
  370 NEXT R
  380 PRINT AT 20,11;Q;" CORRECT "
  390 IF Q=16 AND N=0 THEN GOTO 620
  400 SLOW
  410 LET G$=INKEY$
  420 IF G$="" THEN GOTO 410
  430 IF G$="Z" THEN LET K=X+1
  440 IF G$="Z" THEN LET L=Y
  450 IF G$="X" THEN LET K=X-1
  460 IF G$="X" THEN LET L=Y
  470 IF G$="K" THEN LET K=X
  480 IF G$="K" THEN LET L=Y+1
  490 IF G$="M" THEN LET K=X
  500 IF G$="M" THEN LET L=Y-1
  510 IF G$="Q" THEN GOTO 630
  520 IF G$="P" THEN LET M=-1
  530 LET I=S(K,L)
```

```
540 LET S(K,L)=0
550 LET S(X,Y)=I
560 LET X=K
570 LET Y=L
580 LET M=M+1
590 LET N=0
600 FAST
610 GOTO 260
```

Finally we come to a very simple end routine. You jump to line 330 (*620 ZX81*) if you get the whole square correct and, whether you win or lose, you then get offered another game.

BBC

```
330 PRINT TAB(11,18);"WELL DONE"
340 PRINT TAB(8,20);"PLAY AGAIN Y/N";
350 INPUT A$
360 IF A$="Y" OR A$="YES" THEN RUN
370 STOP
```

RML

```
330 PLOT 22,9,"WELL DONE"
340 PLOT 16,3,"PLAY AGAIN Y/N?"
350 INPUT A$
360 IF A$="Y" OR A$="YES" THEN RUN
370 STOP
```

Spectrum and ZX81

```
330 PRINT AT 18,11;"WELL DONE"
340 PRINT AT 20,8;"PLAY AGAIN Y/N";
350 INPUT A$
360 IF A$="Y" OR A$="YES" THEN RUN
370 STOP
```

Suggestions

Try setting up number squares instead of letters. You could make the game more difficult by increasing the size of the matrix to 5×5.

Full Program: BBC

```
  10 MODE7
  20 VDU23;8202;0;0;0;
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PRINT TAB(6,6);"PUSH ANY KEY TO START"
  50 PRINT TAB(3,10);"SPACE BAR RESETS MOVES TO 0"
  60 PRINT TAB(12,15);"Q = QUIT"
  70 LET G$=INKEY$(3):IF G$="" THEN GOTO 70
  80 CLS
  90 FOR R=1 TO 4
 100 FOR S=1 TO 4
 110 LET S(S,R)=P
 120 LET B(S,R)=P
 130 LET P=P+1
 140 NEXT S:NEXT R
 150 LET S(4,4)=32:LET B(4,4)=32:LET X=4:LET Y=4
 160 FORR=1 TO 4:FOR S=1 TO 4:PRINT TAB(S*6-1,R*4);CHR$
(129+(S(S,R) MOD 7));CHR$(S(S,R)):NEXT S:NEXT R
 170 PRINT TAB(11,1);M;" MOVES  "
 180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
THEN LET Q=Q+1
 190 NEXT S:NEXT R
 200 PRINT TAB(11,20);Q;" CORRECT "
 210 IF Q=16 AND N=0 THEN GOTO 330
 220 *FX15,1
 230 LET G$=INKEY$(3):IF G$="" THEN GOTO 230
 240 IF G$="Z" THEN LET K=X+1:LET L=Y
 250 IF G$="X" THEN LET K=X-1:LET L=Y
 260 IF G$="K" THEN LET K=X:LET L=Y+1
 270 IF G$="M" THEN LET K=X:LET L=Y-1
 280 IF G$="Q" THEN GOTO 340
 290 IF G$=" " THEN LET M=-1
 300 IF S(K,L)=0 THEN GOTO 230
 310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
 320 GOTO 160
 330 PRINT TAB(11,18);"WELL DONE"
 340 PRINT TAB(8,20);"PLAY AGAIN Y/N";
 350 INPUT A$
 360 IF A$="Y" OR A$="YES" THEN RUN
 370 STOP
```

Full Program: Spectrum

```
  10 BORDER 2:PAPER 0:INK 7:CLS
  20 REM
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PRINT AT 6,6;"PUSH ANY KEY TO START"
  50 PRINT AT 10,3;"SPACE BAR RESETS MOVES TO 0"
  60 PRINT AT 15,12;"Q = QUIT"
  70 LET G$=INKEY$:IF G$="" THEN GOTO 70
  80 CLS
  90 FOR R=1 TO 4
 100 FOR S=1 TO 4
 110 LET S(S,R)=P
 120 LET B(S,R)=P
 130 LET P=P+1
 140 NEXT S:NEXT R
 150 LET S(4,4)=32:LET B(4,4)=32:LET X=4:LET Y=4
 160 FORR=1 TO 4:FOR S=1 TO 4:INK R+S:PRINT AT R*4,S*6-
1;CHR$(S(S,R)):NEXT S:NEXT R
 170 PRINT AT 1,11;M;" MOVES  "
 180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
THEN LET Q=Q+1
 190 NEXT S:NEXT R
 200 PRINT AT 20,11;Q;" CORRECT "
 210 IF Q=16 AND N=0 THEN GOTO 330
 220 REM
 230 LET G$=INKEY$:IF G$="" THEN GOTO 230
 240 IF G$="Z" THEN LET K=X+1:LET L=Y
 250 IF G$="X" THEN LET K=X-1:LET L=Y
 260 IF G$="K" THEN LET K=X:LET L=Y+1
 270 IF G$="M" THEN LET K=X:LET L=Y-1
 280 IF G$="Q" THEN GOTO 340
 290 IF G$=" " THEN LET M=-1
 300 IF S(K,L)=0 THEN GOTO 230
 310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
 320 GOTO 160
 330 PRINT AT 18,11;"WELL DONE"
 340 PRINT AT 20,8;"PLAY AGAIN Y/N";
 350 INPUT A$
 360 IF A$="Y" OR A$="YES" THEN RUN
 370 STOP
```

Full Program: RML

```
  10 GRAPH
  20 PUT 12
  30 LET P=65:DIM B(5,5):DIM S(5,5):LET M=0:LET K=0:LET
L=0:LET N=1
  40 PLOT 12,45,"PUSH ANY KEY TO START"
  50 PLOT 6,43,"SPACE BAR RESETS MOVES TO 0"
  60 PLOT 24,18,"Q = QUIT"
  70 LET G$=GET$()
  80 GRAPH
  90 FOR R=1 TO 4
 100 FOR S=1 TO 4
 110 LET S(S,R)=P
 120 LET B(S,R)=P
 130 LET P=P+1
 140 NEXT S:NEXT R
 150 LET S(4,4)=32:LET B(4,4)=32:LET X=4:LET Y=4
 160 FORR=1 TO 4:FOR S=1 TO 4:PLOT S*12-2,(21-R*4)*3,S(
S,R):NEXT S:NEXT R
 170 PUT 12:PRINT TAB(11);M;" MOVES  "
 180 LET Q=0:FOR R=1 TO 4:FOR S=1 TO 4:IF S(S,R)=B(S,R)
THEN LET Q=Q+1
 190 NEXT S:NEXT R
 200 PRINT TAB(10);Q;" CORRECT "
 210 IF Q=16 AND N=0 THEN GOTO 330
 220 REM
 230 LET G$=GET$()
 240 IF G$="Z" THEN LET K=X+1:LET L=Y
 250 IF G$="X" THEN LET K=X-1:LET L=Y
 260 IF G$="K" THEN LET K=X:LET L=Y+1
 270 IF G$="M" THEN LET K=X:LET L=Y-1
 280 IF G$="Q" THEN GOTO 340
 290 IF G$=" " THEN LET M=-1
 300 IF S(K,L)=0 THEN GOTO 230
 310 LET I=S(K,L):LET S(K,L)=32:LET S(X,Y)=I:LET X=K:LE
T Y=L:LET M=M+1:LET N=0
 320 GOTO 160
 330 PLOT 22,9,"WELL DONE"
 340 PLOT 16,3,"PLAY AGAIN Y/N?"
 350 INPUT A$
 360 IF A$="Y" OR A$="YES" THEN RUN
 370 STOP
```

Full Program: ZX81

```
 10 SLOW
 20 LET P=38
 30 DIM B(5,5)
 40 DIM S(5,5)
 50 LET M=0
 60 LET K=0
 70 LET L=0
 80 LET N=1
 90 PRINT AT 6,6;"PRESS ANY KEY TO START"
100 PRINT AT 10,3;"THE P KEY RESETS MOVES TO 0"
110 PRINT AT 15,12;"Q = QUIT"
120 LET G$=INKEY$
130 IF G$="" THEN GOTO 120
140 CLS
150 FAST
160 FOR R=1 TO 4
170 FOR S=1 TO 4
180 LET S(S,R)=P
190 LET B(S,R)=P
200 LET P=P+1
210 NEXT S
215 NEXT R
220 LET S(4,4)=0
230 LET B(4,4)=0
240 LET X=4
250 LET Y=4
260 FOR R=1 TO 4
270 FOR S=1 TO 4
280 PRINT AT R*2,S*6;CHR$ S(S,R)
290 NEXT S
300 NEXT R
310 PRINT AT 1,11;M;" MOVES  "
320 LET Q=0
330 FOR R=1 TO 4
340 FOR S=1 TO 4
350 IF S(S,R)=B(S,R) THEN LET Q=Q+1
360 NEXT S
370 NEXT R
380 PRINT AT 20,11;Q;" CORRECT "
390 IF Q=16 AND N=0 THEN GOTO 620
400 SLOW
410 LET G$=INKEY$
420 IF G$="" THEN GOTO 410
430 IF G$="Z" THEN LET K=X+1
440 IF G$="Z" THEN LET L=Y
450 IF G$="X" THEN LET K=X-1
460 IF G$="X" THEN LET L=Y
470 IF G$="K" THEN LET K=X
480 IF G$="K" THEN LET L=Y+1
490 IF G$="M" THEN LET K=X
500 IF G$="M" THEN LET L=Y-1
510 IF G$="Q" THEN GOTO 630
```

```
520 IF G$="P" THEN LET M=-1
530 LET I=S(K,L)
540 LET S(K,L)=0
550 LET S(X,Y)=I
560 LET X=K
570 LET Y=L
580 LET M=M+1
590 LET N=0
600 FAST
610 GOTO 260
620 PRINT AT 18,11;"WELL DONE"
630 PRINT AT 20,8;"PLAY AGAIN Y/N"
640 INPUT A$
650 IF A$="Y" THEN RUN
660 STOP
```

Pickup

BBC, ZX Spectrum, RML

You are on the run from two relentless bogeymen, who are homing in on you. There is nowhere to hide, no safe place to run to – they are bound to get you in the end. The only thing that will make them pause is if you pick up the O, which the computer places at random on the screen. How many O's can you pick up before the bogeymen catch you?

Player keys

Z – left
X – right
M – down
K – up

The Program

We start by giving your character, a #, coordinates to start from, MX and MY, and the score is set to zero. On the BBC micros we also define the sound effects on lines 10 and 20.

BBC

```
10 ENVELOPE2,1,0,0,0,0,0,0,12,-10,-10,80,-69,127
20 ENVELOPE1,2,0,0,0,0,0,0,60,25,-12,40,50,127
30 ON ERROR GOTO 540
40 MODE7
50 REM
60 *FX11,1
70 *FX12,1
80 LET MX=16:LET MY=11:LET SC=0:LET A$="O"
```

Spectrum

```
10 REM
20 REM
30 REM
40 BORDER 2:PAPER 0:INK 7:CLS
50 REM
60 REM
70 REM
80 LET MX=16:LET MY=11:LET SC=0:LET A$="O"
```

RML

```
 10 GRAPH
 20 RANDOMIZE
 30 REM
 40 PUT 12
 50 REM
 60 REM
 70 REM
 80 LET MX=16:LET MY=11:LET SC=0:LET A$="O"
```

Then we get the screen set up for the start of the game. The two bogeymen are given random coordinates (H1,H3, and H2,H4) in lines 100–130. Line 140 prints your score at the bottom of the screen, line 170 prints your character, and line 160 gives the O you are aiming at random coordinates (A,B).

BBC

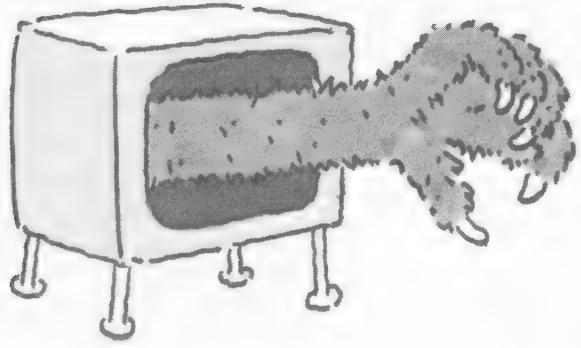
```
  90 PRINTTAB(0,22);CHR$129;CHR$157
 100 LET H1=RND(18)+1:IF H1>MX-3 AND H1<MX+3 THEN GOTO
100
 110 LET H2=RND(18)+1:IF H2>MX-3 AND H2<MX+3 THEN GOTO
110
 120 LET H3=RND(18)+2:IF H3>MY-3 AND H3<MY+3 THEN GOTO
120
 130 LET H4=RND(18)+2:IF H4>MY-3 AND H4<MY+3 THEN GOTO
130
 140 PRINTTAB(0,0);CHR$129;CHR$157;CHR$131;" YOUR SCORE
 IS ";SC
 150 VDU23;8202;0;0;0;
 160 LET A=RND(30)+1:LET B=RND(18)+2
 170 PRINTTAB(MX,MY);"#"
```

Spectrum

```
   90 REM
  100 LET H1=INT(RND*18)+2:IF H1>MX-3 AND H1<MX+3 THEN G
OTO 100
  110 LET H2=INT(RND*18)+2:IF H2>MX-3 AND H2<MX+3 THEN G
OTO 110
  120 LET H3=INT(RND*18)+3:IF H3>MY-3 AND H3<MY+3 THEN G
OTO 120
  130 LET H4=INT(RND*18)+3:IF H4>MY-3 AND H4<MY+3 THEN G
OTO 130
  140 PRINT AT 0,0;" YOUR SCORE IS ";SC
  150 REM
  160 LET A=INT(RND*30)+2:LET B=INT(RND*18)+3
  170 PRINT AT MY,MX;"#"
```

RML

```
   90 FOR R=0 TO 31:PLOT R*2,57,191:NEXT
  100 H1=INT(RND(1)*18)+2:IF H1>MX-3 AND H1<MX+3 THEN GO
TO 100
  110 H2=INT(RND(1)*18)+2:IF H2>MX-3 AND H2<MX+3 THEN GO
TO 110
  120 H3=INT(RND(1)*18)+3:IF H3>MY-3 AND H3<MY+3 THEN GO
TO 120
  130 H4=INT(RND(1)*18)+3:IF H4>MY-3 AND H4<MY+3 THEN GO
TO 120
  140 PRINT"YOUR SCORE IS ";SC
  150 REM
  160 LET A=INT(RND(1)*30)+2:B=INT(RND(1)*18)+3
  170 PLOT MX*2,(21-MY)*3,42
```



The action starts with the main program loop at line 180, which prints the O. The two bogeymen start zooming towards you (lines 190–260), the computer checks whether you are pressing any of the user keys and moves your character accordingly (lines 290–340), and (on the BBC and Spectrum micros) you get a sound which varies as your character moves (line 280). Lines 350–380 prevent your character running off the screen, while lines 400 and 410 check whether the bogeymen have caught you. Finally, the computer checks whether you have picked up the O, before jumping back to the beginning of the main loop again.

BBC

```
 180 PRINTTAB(A,B);A$
 190 PRINTTAB(H1,H3);" "
 200 PRINTTAB(H2,H4);" "
 210 LET H1=H1+SGN(MX-H1)/2
 220 LET H2=H2+SGN(MX-H2)/2
 230 LET H3=H3+SGN(MY-H3)/2
 240 LET H4=H4+SGN(MY-H4)/2
 250 PRINTTAB(H1,H3);"X"
 260 PRINTTAB(H2,H4);"X"
 270 *FX15,1
 280 SOUND1,1,ABS(MX-MY),1
 290 LET G$=INKEY$(4)
 300 PRINTTAB(MX,MY);" "
 310 IF G$="Z" THEN LET MX=MX-1
 320 IF G$="X" THEN LET MX=MX+1
 330 IF G$="M" THEN LET MY=MY+1
 340 IF G$="K" THEN LET MY=MY-1
 350 IF MX<0 THEN LET MX=0
 360 IF MX>31 THEN LET MX=31
 370 IF MY<3 THEN LET MY=3
 380 IF MY>21 THEN LET MY=21
 390 PRINTTAB(MX,MY);"#"
 400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
 410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
 420 IF MX=A AND MY=B THEN GOTO 500
 430 GOTO 180
```

Spectrum

```
180 PRINT AT B,A;A$
190 PRINT AT H3,H1;" "
200 PRINT AT H4,H2;" "
210 LET H1=H1+SGN(MX-H1)/2
220 LET H2=H2+SGN(MX-H2)/2
230 LET H3=H3+SGN(MY-H3)/2
240 LET H4=H4+SGN(MY-H4)/2
250 PRINT AT H3,H1;"X"
260 PRINT AT H4,H2;"X"
270 REM
280 BEEP 0.01,ABS(MX-MY)
290 LET G$=INKEY$
295 IF G$="" THEN GOTO 390
300 PRINT AT MY,MX;" "
310 IF G$="Z" THEN LET MX=MX-1
320 IF G$="X" THEN LET MX=MX+1
330 IF G$="M" THEN LET MY=MY+1
340 IF G$="K" THEN LET MY=MY-1
350 IF MX<0 THEN LET MX=0
360 IF MX>31 THEN LET MX=31
370 IF MY<3 THEN LET MY=3
380 IF MY>21 THEN LET MY=21
390 PRINT AT MY,MX;"#"
400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
420 IF MX=A AND MY=B THEN GOTO 500
430 GOTO 180
```

RML

```
180 PLOT A*2,(21-B)*3,A$
190 PLOT H1*2,(21-H3)*3,32
200 PLOT H2*2,(21-H4)*3,32
210 LET H1=H1+SGN(MX-H1)/2
220 LET H2=H2+SGN(MX-H2)/2
230 LET H3=H3+SGN(MY-H3)/2
240 LET H4=H4+SGN(MY-H4)/2
250 PLOT H1*2,(21-H3)*3,"X"
260 PLOT H2*2,(21-H4)*3,"X"
270 REM
280 REM
290 G$=GET$(4)
300 PLOT MX*2,(21-MY)*3,32
310 IF G$="Z" THEN LET MX=MX-1
320 IF G$="X" THEN LET MX=MX+1
330 IF G$="/" THEN LET MY=MY+1
340 IF G$=":" THEN LET MY=MY-1
350 IF MX<0 THEN LET MX=0
360 IF MX>31 THEN LET MX=31
```

```
370 IF MY<3 THEN LET MY=3
380 IF MY>21 THEN LET MY=21
390 PLOT MX*2,(21-MY)*3,42
400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
420 IF MX=A AND MY=B THEN GOTO 500
430 GOTO 180
```

When the bogeymen catch you, they're laughing! Lines 400 and 410 send you to this lose routine, which prints 'HA!' all over the screen before running the program again.

BBC

```
440 FORR=1 TO 60
450 PRINTTAB(RND(36),RND(20));CHR$(RND(7)+128);"HA!"
460 NEXT R
470 *FX15,1
480 FORR=1 TO 5000:NEXT R
490 RUN
```

Spectrum

```
440 FORR=1 TO 60
450 PRINT AT RND*20,RND*29;"HA!"
460 NEXT R
470 REM
480 FORR=1 TO 500:NEXT R
490 RUN
```

RML

```
440 FORR=1 TO 60
450 PLOT MX*2,(21-MY)*3,195+R
460 NEXT R
470 REM
480 FORR=1 TO 5000:NEXT R
490 RUN
```

However, when you do manage to pick up the O, the bogeymen have to start all over again, and you get one added to your score (line 500) before the next round.

BBC

```
500 LET SC=SC+1
510 FORR=1 TO 3:SOUND2,2,155,3:NEXT R
520 CLS
530 GOTO 90
540 *FX11,66
550 *FX12,4
560 STOP
```

Spectrum

```
500 LET SC=SC+1
510 BEEP 0.5,35
520 CLS
530 GOTO 90
```

RML

```
500 SC=SC+1
510 REM
520 PUT 12
530 GOTO 90
```

Suggestions

Try making the bogeymen home in on you a bit faster by changing from ½ steps to 1½ steps in lines 210–240.

Full Program: BBC

```
  10 ENVELOPE2,1,0,0,0,0,0,0,0,12,-10,-10,80,-69,127
  20 ENVELOPE1,2,0,0,0,0,0,0,0,60,25,-12,40,50,127
  30 ON ERROR GOTO 540
  40 MODE7
  50 REM
  60 *FX11,1
  70 *FX12,1
  80 LET MX=16:LET MY=11:LET SC=0:LET A$="0"
  90 PRINTTAB(0,22);CHR$129;CHR$157
 100 LET H1=RND(18)+1:IF H1>MX-3 AND H1<MX+3 THEN GOTO
100
 110 LET H2=RND(18)+1:IF H2>MX-3 AND H2<MX+3 THEN GOTO
110
 120 LET H3=RND(18)+2:IF H3>MY-3 AND H3<MY+3 THEN GOTO
120
 130 LET H4=RND(18)+2:IF H4>MY-3 AND H4<MY+3 THEN GOTO
130
 140 PRINTTAB(0,0);CHR$129;CHR$157;CHR$131;" YOUR SCORE
 IS ";SC
 150 VDU23;8202;0;0;0;
 160 LET A=RND(30)+1:LET B=RND(18)+2
 170 PRINTTAB(MX,MY);"#"
 180 PRINTTAB(A,B);A$
 190 PRINTTAB(H1,H3);" "
 200 PRINTTAB(H2,H4);" "
 210 LET H1=H1+SGN(MX-H1)/2
 220 LET H2=H2+SGN(MX-H2)/2
 230 LET H3=H3+SGN(MY-H3)/2
 240 LET H4=H4+SGN(MY-H4)/2
 250 PRINTTAB(H1,H3);"X"
 260 PRINTTAB(H2,H4);"X"
 270 *FX15,1
 280 SOUND1,1,ABS(MX-MY),1
 290 LET G$=INKEY$(4)
 300 PRINTTAB(MX,MY);" "
 310 IF G$="Z" THEN LET MX=MX-1
 320 IF G$="X" THEN LET MX=MX+1
 330 IF G$="M" THEN LET MY=MY+1
 340 IF G$="K" THEN LET MY=MY-1
 350 IF MX<0 THEN LET MX=0
 360 IF MX>31 THEN LET MX=31
 370 IF MY<3 THEN LET MY=3
 380 IF MY>21 THEN LET MY=21
 390 PRINTTAB(MX,MY);"#"
 400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
 410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
 420 IF MX=A AND MY=B THEN GOTO 500
 430 GOTO 180
 440 FORR=1 TO 60
 450 PRINTTAB(RND(36),RND(20));CHR$(RND(7)+128);"HA!"
```

```
460 NEXT R
470 *FX15,1
480 FORR=1 TO 5000:NEXT R
490 RUN
500 LET SC=SC+1
510 FORR=1 TO 3:SOUND2,2,155,3:NEXT R
520 CLS
530 GOTO 90
540 *FX11,66
550 *FX12,4
560 STOP
```

Full Program: Spectrum

```
 10 REM
 20 REM
 30 REM
 40 BORDER 2:PAPER 0:INK 7:CLS
 50 REM
 60 REM
 70 REM
 80 LET MX=16:LET MY=11:LET SC=0:LET A$="0"
 90 REM
100 LET H1=INT(RND*18)+2:IF H1>MX-3 AND H1<MX+3 THEN G
OTO 100
110 LET H2=INT(RND*18)+2:IF H2>MX-3 AND H2<MX+3 THEN G
OTO 110
120 LET H3=INT(RND*18)+3:IF H3>MY-3 AND H3<MY+3 THEN G
OTO 120
130 LET H4=INT(RND*18)+3:IF H4>MY-3 AND H4<MY+3 THEN G
OTO 130
140 PRINT AT 0,0;" YOUR SCORE IS ";SC
150 REM
160 LET A=INT(RND*30)+2:LET B=INT(RND*18)+3
170 PRINT AT MY,MX;"#"
180 PRINT AT B,A;A$
190 PRINT AT H3,H1;" "
200 PRINT AT H4,H2;" "
210 LET H1=H1+SGN(MX-H1)/2
220 LET H2=H2+SGN(MX-H2)/2
230 LET H3=H3+SGN(MY-H3)/2
240 LET H4=H4+SGN(MY-H4)/2
250 PRINT AT H3,H1;"X"
260 PRINT AT H4,H2;"X"
270 REM
280 BEEP 0.01,ABS(MX-MY)
290 LET G$=INKEY$
295 IF G$="" THEN GOTO 390
300 PRINT AT MY,MX;" "
310 IF G$="Z" THEN LET MX=MX-1
```

```
 320 IF G$="X" THEN LET MX=MX+1
 330 IF G$="M" THEN LET MY=MY+1
 340 IF G$="K" THEN LET MY=MY-1
 350 IF MX<0 THEN LET MX=0
 360 IF MX>31 THEN LET MX=31
 370 IF MY<3 THEN LET MY=3
 380 IF MY>21 THEN LET MY=21
 390 PRINT AT MY,MX;"#"
 400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
 410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
 420 IF MX=A AND MY=B THEN GOTO 500
 430 GOTO 180
 440 FORR=1 TO 60
 450 PRINT AT RND*20,RND*29;"HA!"
 460 NEXT R
 470 REM
 480 FORR=1 TO 500:NEXT R
 490 RUN
 500 LET SC=SC+1
 510 BEEP 0.5,35
 520 CLS
 530 GOTO 90
```

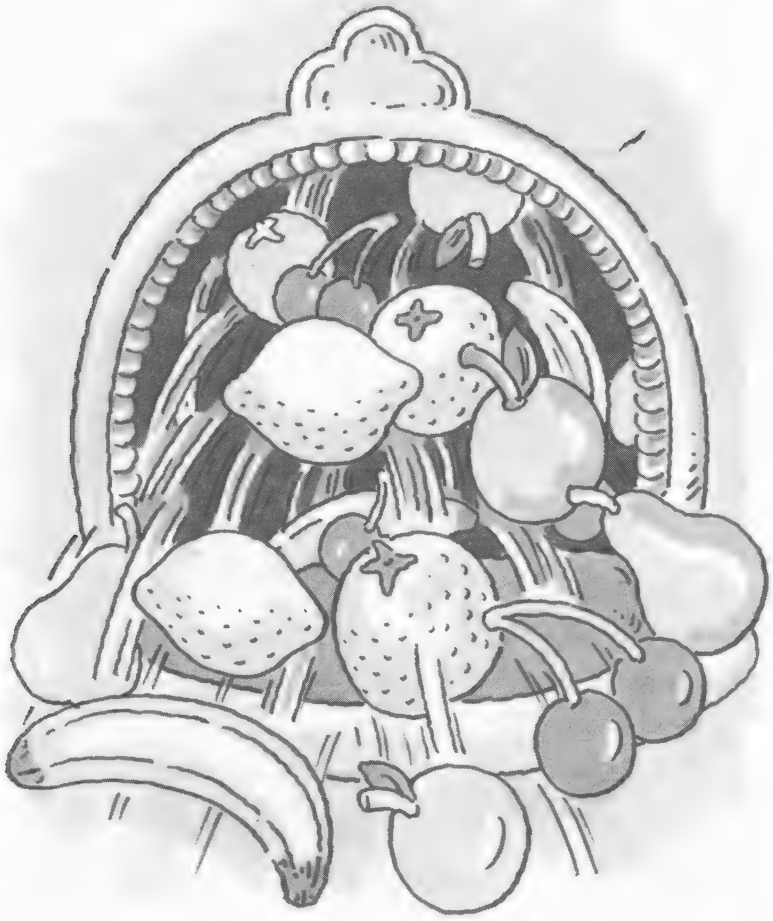
Full Program: RML

```
  10 GRAPH
  20 RANDOMIZE
  30 REM
  40 PUT 12
  50 REM
  60 REM
  70 REM
  80 LET MX=16:LET MY=11:LET SC=0:LET A$="O"
  90 FOR R=0 TO 31:PLOT R*2,57,191:NEXT
 100 H1=INT(RND(1)*18)+2:IF H1>MX-3 AND H1<MX+3 THEN GO
TO 100
 110 H2=INT(RND(1)*18)+2:IF H2>MX-3 AND H2<MX+3 THEN GO
TO 110
 120 H3=INT(RND(1)*18)+3:IF H3>MY-3 AND H3<MY+3 THEN GO
TO 120
 130 H4=INT(RND(1)*18)+3:IF H4>MY-3 AND H4<MY+3 THEN GO
TO 120
 140 PRINT"YOUR SCORE IS ";SC
 150 REM
 160 LET A=INT(RND(1)*30)+2:B=INT(RND(1)*18)+3
 170 PLOT MX*2,(21-MY)*3,42
 180 PLOT A*2,(21-B)*3,A$
 190 PLOT H1*2,(21-H3)*3,32
```

```
 200 PLOT H2*2,(21-H4)*3,32
 210 LET H1=H1+SGN(MX-H1)/2
 220 LET H2=H2+SGN(MX-H2)/2
 230 LET H3=H3+SGN(MY-H3)/2
 240 LET H4=H4+SGN(MY-H4)/2
 250 PLOT H1*2,(21-H3)*3,"X"
 260 PLOT H2*2,(21-H4)*3,"X"
 270 REM
 280 REM
 290 G$=GET$(4)
 300 PLOT MX*2,(21-MY)*3,32
 310 IF G$="Z" THEN LET MX=MX-1
 320 IF G$="X" THEN LET MX=MX+1
 330 IF G$="/" THEN LET MY=MY+1
 340 IF G$=":" THEN LET MY=MY-1
 350 IF MX<0 THEN LET MX=0
 360 IF MX>31 THEN LET MX=31
 370 IF MY<3 THEN LET MY=3
 380 IF MY>21 THEN LET MY=21
 390 PLOT MX*2,(21-MY)*3,42
 400 IF H1>=MX-1 AND H1<=MX+1 AND H3>=MY-1 AND H3<=MY+1
THEN GOTO 440
 410 IF H2>=MX-1 AND H2<=MX+1 AND H4>=MY-1 AND H4<=MY+1
THEN GOTO 440
 420 IF MX=A AND MY=B THEN GOTO 500
 430 GOTO 180
 440 FORR=1 TO 60
 450 PLOT MX*2,(21-MY)*3,195+R
 460 NEXT R
 470 REM
 480 FORR=1 TO 5000:NEXT R
 490 RUN
 500 SC=SC+1
 510 REM
 520 PUT 12
 530 GOTO 90
```

Fruit Machine

BBC, ZX Spectrum, RML, ZX81



This game is for anyone who likes playing with one-armed bandits but hates the way the machines swallow up your money. Here is your own home fruit machine, complete with nudge controls, just waiting to challenge your luck!

All you have to do to win is to get two or more of the same characters together, reading from the left. For example,

A A D C

would score a win for the two A's, but

C D A A

would not score anything.

Assuming that each go costs you 10p, the computer will work out your winnings for you and tell you how much profit you have made (if any).

Player keys

Any key will spin the reels.
When you get a chance to 'nudge', use 1, 2, 3 and 4 to tell the computer which reel to nudge.

The programs for Spectrum and ZX81 are so similar that we are only giving you the listing for ZX81, with the Spectrum variations listed below. The lines you need to change for the Spectrum are marked with a S. *The BBC fruit machine is more complicated than the others and so is explained separately..*

The Program: Spectrum, RML, ZX81

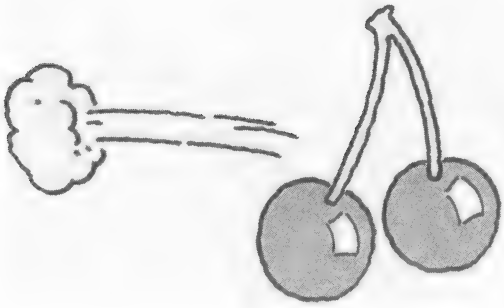
The first thing to do is to print up the fruit machine. Lines 10–100 print bars across the screen, leaving three lines blank in the middle for the reels. The Spectrum and the ZX81 micros use the graphic ▄ in line 30 and ▀ in line 90, both for 31 character spaces across the screen. Line 110 sets up an array (A) of seven elements into which seven random letters will be put for the fruit machine reels. Line 120 sets your current profit (s) to zero.

Spectrum and ZX81

```
 10 RAND
 15 INK 0:PAPER 7:CLS
 20 FOR T=1 TO 8
 30 PRINT"                                "
 40 NEXT T
 50 PRINT
 60 PRINT
 70 PRINT
 80 FOR T=1 TO 10
 90 PRINT"                                "
100 NEXT T
110 DIM A(7)
120 LET S=0
```

RML

```
  5 GRAPH
 10 RANDOMIZE:PUT 12
 20 FOR T=1 TO 9
 30 FOR R=1 TO 31:PLOT R*2,(21-T)*3,204:NEXT R
 40 NEXT T
 50 REM
 60 REM
 70 REM
 80 FOR T=13 TO 21
 90 FOR R=1 TO 31:PLOT R*2,(21-T)*3,204:NEXT R
100 NEXT T
110 DIM A(7):A$=CHR$(204)+CHR$(204)+CHR$(204)+CHR$(204
)+CHR$(204):B$=A$+A$
120 LET S=0
```



Now for the main loop. Lines 140–170 choose four random letters from the first seven of the alphabet, which are then printed up in lines 200–220. You are given a one in six chance of getting a nudge (lines 230–240), and if you're lucky the program jumps to the nudge subroutine at line 390. The computer checks whether you have a possible winning combination (lines 250–270), works out your winnings by jumping to a subroutine at line 350, and works out your profit so far, after deducting a hypothetical 10p for your go (line 280). Both the winnings and profit are flashed up for you (line 300 and line 350).

ZX81

```
S 130 FAST
  140 FOR R=1 TO 4
S 150 LET C=INT(RND*6)+38
  160 LET A(R)=C
  170 NEXT R
  180 LET W=0
  190 PRINT
  200 FOR R=1 TO 4
  210 PRINT AT 9,6*R;CHR$(A(R))
  220 NEXT R
  230 LET T=INT(RND*6)
  240 IF T=1 THEN GOTO 390
  250 IF A(1)=A(2) THEN LET W=20
  260 IF A(1)=A(2) AND A(2)=A(3) THEN LET W=100
  270 IF W=100 AND A(3)=A(4) THEN LET W=500
  280 LET S=S-10
  290 IF W>0 THEN GOTO 350
  300 PRINT AT 15,10;" PROFIT=";S;"P■"
S 310 SLOW
  320 IF INKEY$="" THEN GOTO 320
  330 PRINT AT 4,12;"■■■■■■■■■■"
  340 GOTO 130
  350 PRINT AT 4,12;" WIN=";W;" P"
  360 LET S=S+W
  370 LET W=0
  380 GOTO 300
```

Spectrum changes and additions

```
 15 INK 0:PAPER 7:CLS
130 BORDER 6
150 LET C=INT(RND*6)+65
205 INK A(R)-64
225 INK 0
310 REM
355 BEEP 0.5,38
```

RML

```
130 REM
140 FOR R=1 TO 4
150 LET C=INT(RND(1)*6)+65
160 LET A(R)=C
170 NEXT R
180 LET W=0
190 REM
200 FOR R=1 TO 4
210 PLOT 12*R,30,A(R)
220 NEXT R
230 LET T=INT(RND(1)*6)+1
240 IF T=1 THEN GOTO 390
250 IF A(1)=A(2) THEN LET W=20
260 IF A(1)=A(2) AND A(2)=A(3) THEN LET W=100
270 IF W=100 AND A(3)=A(4) THEN LET W=500
280 LET S=S-10:PUT 12
290 IF W>0 THEN GOTO 350
300 PRINT TAB(10);"PROFIT=";S;"P "
310 REM
320 G=GET()
330 REM
340 GOTO 130
350 PRINT TAB(11);"WIN=";W;"P "
360 LET S=S+W
370 LET W=0
380 GOTO 300
```

Then we come to the nudge routine, to which you are sent if you strike lucky in line 240. Lines 390–450 print a flashing 'NUDGE NOW' at the top of the screen, which is achieved by alternating the letters with a black bar – graphic ▄ for ten character spaces on the ZX81 and the Spectrum – using a little delay loop in lines 670–690. The computer tells you how many

nudges it will allow you (lines 460–490), and then checks which key you are pressing (lines 500–560). The corresponding reel changes to a new character (lines 57–610) and this continues until you have used up all your nudges. The computer then jumps back into the main loop at line 250 to work out your winnings.

ZX81

```
S 390 SLOW
  400 FOR H=1 TO 4
  410 PRINT AT 2,11;"■■■■■■■■■■"
  420 GOSUB 670
  430 PRINT AT 2,11;"NUDGE NOW "
  440 GOSUB 670
  450 NEXT H
  460 LET E=INT(RND*5)+2
  470 FOR R=1 TO E
  480 LET B=0
  490 PRINT AT 3,12;E-R+1;" NUDGES"
  500 LET G$=INKEY$
  510 IF G$="" THEN GOTO 500
  520 IF G$="1" THEN LET B=1
  530 IF G$="2" THEN LET B=2
  540 IF G$="3" THEN LET B=3
  550 IF G$="4" THEN LET B=4
  560 IF B=0 THEN GOTO 620
  570 LET Z=A(B)
S 580 LET Y=INT(RND*6)+38
  590 IF Y=Z THEN GOTO 580
  600 LET A(B)=Y
  610 PRINT AT 9,6*B;CHR$(A(B))
  620 NEXT R
S 630 FAST
  640 PRINT AT 2,11;"■■■■■■■■■■"
  650 PRINT AT 3,12;"■■■■■■■■"
  660 GOTO 250
  670 FOR R=1 TO 4
  680 NEXT R
  690 RETURN
```

Spectrum changes

```
390 REM
425 INK 2
435 INK 0
580 LET Y=INT(RND*6)+65
605 INK Y-64
615 INK 0
630 REM
670 FOR R=1 TO 20
```

RML

```
390 REM
400 FOR H=1 TO 4
410 PLOT 22,57,B$
420 GOSUB 670
430 PLOT 22,57,"NUDGE NOW"
440 GOSUB 670
450 NEXT H
460 LET E=INT(RND(1)*5)+3
470 FOR R=1 TO E
480 LET B=0
490 PUT 12:PRINT TAB(8);"YOU HAVE ";E-R+1;" NUDGES"
500 G$=GET$()
510 REM
520 IF G$="1" THEN LET B=1
530 IF G$="2" THEN LET B=2
540 IF G$="3" THEN LET B=3
550 IF G$="4" THEN LET B=4
560 IF B=0 THEN GOTO 620
570 LET Z=A(B)
580 LET Y=INT(RND(1)*6)+65
590 IF Y=Z THEN GOTO 580
600 LET A(B)=Y
610 PLOT 12*B,30,A(B)
620 NEXT R
630 REM
640 PLOT 22,57,B$
650 REM
660 GOTO 250
670 FORR=1 TO 200
680 NEXT R
690 RETURN
```

The Program: BBC (Model B only)

This program has been written in Mode 2 for the range of colours available. Model A users can run it in Mode 5, but will have to change the colour statements accordingly. First of all, we will draw seven characters for our fruit machine. These are a bell, a cherry, a £ sign, an alien, QU, II and ◇.

```
 10 MODE2
 20 COLOUR1:PRINTTAB(3,3)"FRUIT MACHINE"
 30 VDU 23,255,28,54,48,124,48,48,126,0
 40 VDU 23,249,0,254,108,108,108,108,254,0
 50 VDU 23,250,245,149,149,149,149,165,215,0
 60 VDU 23,251,24,60,126,126,126,126,255,24
 70 VDU 23,252,145,56,124,254,124,56,145,0
 80 VDU 23,253,7,12,56,124,254,254,124,56
 90 VDU 23,254,102,153,60,90,255,129,126,36
```

Then we arrange for some simple instructions. Some of the spacing in the instructions may look a bit odd. This is not a mistake. It is just to avoid words being broken at the end of the line. Line 160 checks whether you are pressing a key to start the reels spinning.

```
 100 PRINTTAB(3,4)"-------------"
 110 COLOUR3:PRINTTAB(0,7)"Push any key to spin       r
eels"
 120 COLOUR2:PRINTTAB(0,10)"On 'NUDGE NOW' push either
1,2,3 or 4,  according to which  reel you want to    nud
ge."
 130 COLOUR5:PRINTTAB(0,16)"On 'HOLD NOW' push  either
1,2,3 or 4,  according to which  reel you want to    hol
d."
 140 COLOUR6:PRINTTAB(0,21)"      ========="
 150 COLOUR4:PRINTTAB(0,23)"If you win more than'10 the
 game ends."
 160 F=GET
```

Lines 170 and 180 produce the sound effects and lines 190 and 200 turn off the auto-repeat delay altogether.

```
 170 ENVELOPE1,1,-5,0,5,50,50,50,127,0,0,-127,126,0
 180 ENVELOPE2,1,-5,-2,9,15,5,10,127,0,0,-127,126,0
 190 *FX12,0
 200 *FX11,0
 210 ON ERROR GOTO 750
```

Line 230 sets up the dimensions for arrays A and H, into which the characters will be placed. It also gives initial data for the computer to work from. GO is your total expenditure, TOT is your winnings. JK is a flag for the hold routine and GH is a flag for the nudge routine. Line 240 turns off the cursor and lines 250–280 draw the fruit machine.

```
  220 REM *****FRUIT MACHINE****
  230 MODE2:DIMA(6):DIMH(6):GO=0:ToT=0:A$="":JK=0:GH=0
  240 VDU23;8202;0;0;0;
  250 MOVE1280,0:MOVE1280,420:GCOL0,4:PLOT 85,0,0:PLOT 8
5,0,420
  260 MOVE1280,1024:MOVE1280,604:PLOT85,0,1024:PLOT 85,0
,604
  270 MOVE 0,0:GCOL0,3:DRAW0,1023:DRAW 1279,1023:DRAW 12
79,0:DRAW 0,0
  280 GCOL0,1:MOVE 16,424:DRAW 1263,424:MOVE16,596:DRAW
 1263,596:GCOL0,4
```

Then we come to the main loop of the program. Lines 290–400 spin the reels, making the characters appear in rapid succession, in a variety of colours. You are given a one in five chance for a hold (line 600) and a one in five chance for a nudge (line 390). The computer then checks whether you have won anything (lines 420–480) and prints up your totals (lines 520–570). If you win more than £10, you get a special chord in recognition (line 580).

```
  290 FORE=1 TO 19
  300 FORR=1 TO 4:IF H(R)=10 THEN NEXT ELSE A(R)=RND(7):
NEXT
  310 H=16
  320 FORT=1 TO 4
  330 COLOURA(T):PRINTTAB(4*T,H)CHR$(A(T)+248)
  340 NEXT
  350 NEXT
  360 FORR=1 TO 5:H(R)=0:NEXT:COLOUR132:PRINTTAB(4,H-4)"
 ":PRINTTAB(16,H-4)" ":PRINTTAB(12,H-4)" ":PRINTTAB(6,H-
4)"          "
  370 PRINTTAB(2,6)"                    "
  380 IF JK=12 OR GH=12 THEN 400
  390 G=RND(5):IF G=1 THEN SOUND1,1,180,30:PROCnudge ELS
E GH=0
  400 A$="          ":JK=0
  410 GO=GO+10
```

```
  420 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)=
7 THEN A$="£7.00":ToT=ToT+700:GOTO 500
  430 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)>
4 THEN A$="£5.00":ToT=ToT+500:GOTO 500
  440 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)<
=4 THEN A$="£3.00":ToT=ToT+300:GOTO 500
  450 IF A(1)=A(2) AND A(2)=A(3) THEN A$="£1.00":ToT=ToT
+100:GOTO 500
  460 IF A(1)=A(2) AND A(1)=7 THEN A$="£0.70":ToT=ToT+70
:GOTO 500
  470 IF A(1)=A(2) AND A(1)>3 THEN A$="£0.50":ToT=ToT+50
:GOTO 500
  480 IF A(1)=A(2) AND A(1)<=3 THEN A$="£0.20":ToT=ToT+2
0:GOTO 500
  490 GOTO 520
  500 SOUND1,-12,125,4
  510 GH=12
  520 COLOUR132:COLOUR7
  530 IF A$<>"" PRINTTAB(7,12)A$
  540 PRINTTAB(1,20)"TOTAL WINS=";ToT;" p."
  550 PRINTTAB(1,22)"CASH SPENT=";GO;" p."
  560 COLOUR5:PRINTTAB(4,27)"PROFIT=";(ToT-GO);"p.   "
  570 COLOUR128
  580 IF ToT-GO>1000 THEN COLOUR1:COLOUR131:PRINTTAB(0,H
)"****** YOU WIN *****":PROCchord:GOTO 750
  590 IF GH=12 THEN GH=0:GOTO 610
  600 W=RND(5):IF W=1 THEN SOUND2,2,200,30:PROChold:JK=1
2:GOTO 290
  610 G=GET:GOTO 290
```

If you are lucky at line 390, you are sent to this nudge procedure which tells you how many nudges you have (line 660), checks which key you are pressing (line 680) and moves the relevant reel (700). When you come to the end of your nudges, or fail to nudge at all, the computer jumps back to its place in the main loop.

```
  620 DEFPROCnudge
  630 COLOUR11:COLOUR132:PRINTTAB(3,5)"**NUDGE NOW**"
  640 GH=12:V=RND(5)+2
  650 FORY=1 TO V
  660 COLOUR132:COLOUR12:PRINTTAB(1,7)"YOU HAVE ";V;" NU
DGES"
  670 V=V-1
  680 S=GET:S=S-48:L=1:IF S<1 OR S>4 THEN 720
  690 EW=RND(7):IF EW=A(S) THEN GOTO 690 ELSE A(S)=EW
  700 COLOUR128:FORT=1 TO 4:COLOURA(T):PRINTTAB(4*T,H)CH
R$(A(T)+248):NEXT
  710 NEXT
  720 COLOUR132
  730 MOVE1271,1019:MOVE1271,604:PLOT85,8,1019:PLOT 85,8
,604
  740 ENDPROC
```

When you get a chance to hold a reel (line 660), you are sent to this procedure at line 800, which checks which key you are pressing (line 840) and prints H above your chosen reel before jumping back to the main loop.

```
  800 DEFPROChold
  810 COLOUR132:COLOUR11:PRINTTAB(2,6)"  **";:COLOUR0:PR
INT"HOLD NOW";:COLOUR12:PRINT"**  "
  820 COLOUR8
  830 RF=1
  840 Z=GET:Z=Z-48:IF Z<1 OR Z>4 THEN Z=6
  850 IF Z=6 THEN COLOUR128:ENDPROC
  860 H(Z)=10:PRINTTAB((4*Z),H-4)"H"
  870 RF=RF+1:IF RF>4 THEN COLOUR128:ENDPROC ELSE GOTO 8
40
```

If your winnings reach £10, the computer plays a special chord and then returns to the main loop.

```
  880 DEFPROCchord
  890 SOUND1,-10,40,50:SOUND2,-10,60,50:SOUND3,-10,76,50
:ENDPROC
```

Finally, when you have reached the £10 goal, the computer offers you another game.

```
  750 *FX12,4
  760 *FX11,55
  770 COLOUR7:COLOUR132
  780 PRINT'''" PLAY AGAIN ? (Y/N) ":G$=GET$:IF G$="Y" T
HEN RUN ELSE IF G$="N" THEN CLS:END
  790 GOTO 780
```

Full Program: BBC

```
   10 MODE2
   20 COLOUR1:PRINTTAB(3,3)"FRUIT MACHINE"
   30 VDU 23,255,28,54,48,124,48,48,126,0
   40 VDU 23,249,0,254,108,108,108,108,254,0
   50 VDU 23,250,245,149,149,149,149,165,215,0
   60 VDU 23,251,24,60,126,126,126,126,255,24
   70 VDU 23,252,145,56,124,254,124,56,145,0
   80 VDU 23,253,7,12,56,124,254,254,124,56
   90 VDU 23,254,102,153,60,90,255,129,126,36
  100 PRINTTAB(3,4)"-------------"
  110 COLOUR3:PRINTTAB(0,7)"Push any key to spin       r
eels"
  120 COLOUR2:PRINTTAB(0,10)"On 'NUDGE NOW' push either
1,2,3 or 4,  according to which  reel you want to    nud
ge."
  130 COLOUR5:PRINTTAB(0,16)"On 'HOLD NOW' push  either
1,2,3 or 4,  according to which  reel you want to    hol
d."
  140 COLOUR6:PRINTTAB(0,21)"       =========="
  150 COLOUR4:PRINTTAB(0,23)"If you win more than£10 the
 game ends."
  160 F=GET
  170 ENVELOPE1,1,-5,0,5,50,50,50,127,0,0,-127,126,0
  180 ENVELOPE2,1,-5,-2,9,15,5,10,127,0,0,-127,126,0
  190 *FX12,0
  200 *FX11,0
  210 ON ERROR GOTO 750
  220 REM *****FRUIT MACHINE****
  230 MODE2:DIMA(6):DIMH(6):GO=0:ToT=0:A$="":JK=0:GH=0
  240 VDU23;8202;0;0;0;
  250 MOVE1280,0:MOVE1280,420:GCOL0,4:PLOT 85,0,0:PLOT 8
5,0,420
  260 MOVE1280,1024:MOVE1280,604:PLOT85,0,1024:PLOT 85,0
,604
  270 MOVE 0,0:GCOL0,3:DRAW0,1023:DRAW 1279,1023:DRAW 12
79,0:DRAW 0,0
  280 GCOL0,1:MOVE 16,424:DRAW 1263,424:MOVE 16,596:DRAW
 1263,596:GCOL0,4
```

```
  290 FORE=1 TO 19
  300 FORR=1 TO 4:IF H(R)=10 THEN NEXT ELSE A(R)=RND(7):
NEXT
  310 H=16
  320 FORT=1 TO 4
  330 COLOURA(T):PRINTTAB(4*T,H)CHR$(A(T)+248)
  340 NEXT
  350 NEXT
  360 FORR=1 TO 5:H(R)=0:NEXT:COLOUR132:PRINTTAB(4,H-4)"
 ":PRINTTAB(16,H-4)" ":PRINTTAB(12,H-4)" ":PRINTTAB(6,H-
4)"              "
  370 PRINTTAB(2,6)"                  "
  380 IF JK=12 OR GH=12 THEN 400
  390 G=RND(5):IF G=1 THEN SOUND1,1,180,30:PROCnudge ELS
E GH=0
  400 A$="           ":JK=0
  410 GO=GO+10
  420 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)=
7 THEN A$="£7.00":ToT=ToT+700:GOTO 500
  430 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)>
4 THEN A$="£5.00":ToT=ToT+500:GOTO 500
  440 IF A(1)=A(2) AND A(2)=A(3) AND A(3)=A(4) AND A(4)<
=4 THEN A$="£3.00":ToT=ToT+300:GOTO 500
  450 IF A(1)=A(2) AND A(2)=A(3) THEN A$="£1.00":ToT=ToT
+100:GOTO 500
  460 IF A(1)=A(2) AND A(1)=7 THEN A$="£0.70":ToT=ToT+70
:GOTO 500
  470 IF A(1)=A(2) AND A(1)>3 THEN A$="£0.50":ToT=ToT+50
:GOTO 500
  480 IF A(1)=A(2) AND A(1)<=3 THEN A$="£0.20":ToT=ToT+2
0:GOTO 500
  490 GOTO 520
  500 SOUND1,-12,125,4
  510 GH=12
  520 COLOUR132:COLOUR7
  530 IF A$<>"" PRINTTAB(7,12)A$
  540 PRINTTAB(1,20)"TOTAL WINS=";ToT;" p."
  550 PRINTTAB(1,22)"CASH SPENT=";GO;" p."
  560 COLOUR5:PRINTTAB(4,27)"PROFIT=";(ToT-GO);"p.   "
  570 COLOUR128
  580 IF ToT-GO>1000 THEN COLOUR1:COLOUR131:PRINTTAB(0,H
)"****** YOU WIN *****":PROCchord:GOTO 750
  590 IF GH=12 THEN GH=0:GOTO 610
  600 W=RND(5):IF W=1 THEN SOUND2,2,200,30:PROChold:JK=1
2:GOTO 290
  610 G=GET:GOTO 290
  620 DEFPROCnudge
  630 COLOUR11:COLOUR132:PRINTTAB(3,5)"**NUDGE NOW**"
  640 GH=12:V=RND(5)+2
  650 FORY=1 TO V
  660 COLOUR132:COLOUR12:PRINTTAB(1,7)"YOU HAVE ";V;" NU
DGES"
  670 V=V-1
  680 S=GET:S=S-48:L=1:IF S<1 OR S>4 THEN 720
```

```
  690 EW=RND(7):IF EW=A(S) THEN GOTO 690 ELSE A(S)=EW
  700 COLOUR128:FORT=1 TO 4:COLOURA(T):PRINTTAB(4*T,H)CH
R$(A(T)+248):NEXT
  710 NEXT
  720 COLOUR132
  730 MOVE1271,1019:MOVE1271,604:PLOT85,8,1019:PLOT 85,8
,604
  740 ENDPROC
  750 *FX12,4
  760 *FX11,55
  770 COLOUR7:COLOUR132
  780 PRINT'''" PLAY AGAIN ? (Y/N) ":G$=GET$:IF G$="Y" T
HEN RUN ELSE IF G$="N" THEN CLS:END
  790 GOTO 780
  800 DEFPROChold
  810 COLOUR132:COLOUR11:PRINTTAB(2,6)"  **";:COLOUR0:PR
INT"HOLD NOW";:COLOUR12:PRINT"**  "
  820 COLOUR8
  830 RF=1
  840 Z=GET:Z=Z-48:IF Z<1 OR Z>4 THEN Z=6
  850 IF Z=6 THEN COLOUR128:ENDPROC
  860 H(Z)=10:PRINTTAB((4*Z),H-4)"H"
  870 RF=RF+1:IF RF>4 THEN COLOUR128:ENDPROC ELSE GOTO 8
40
  880 DEFPROCchord
  890 SOUND1,-10,40,50:SOUND2,-10,60,50:SOUND3,-10,76,50
:ENDPROC
```

Full Programme: Spectrum

```
 10 RAND
 15 INK 0:PAPER 7:CLS
 20 FOR T=1 TO 8
 30 PRINT"                                "
 40 NEXT T
 50 PRINT
 60 PRINT
 70 PRINT
 80 FOR T=1 TO 10
 90 PRINT"                                "
100 NEXT T
110 DIM A(7)
120 LET S=0
130 BORDER 6
140 FOR R=1 TO 4
150 LET C=INT(RND*6)+65
160 LET A(R)=C
170 NEXT R
```

```
180 LET W=0
190 PRINT
200 FOR R=1 TO 4
205 INK A(R)-64
210 PRINT AT 9,6*R;CHR$(A(R))
220 NEXT R
225 INK 0
230 LET T=INT(RND*6)
240 IF T=1 THEN GOTO 390
250 IF A(1)=A(2) THEN LET W=20
260 IF A(1)=A(2) AND A(2)=A(3) THEN LET W=100
270 IF W=100 AND A(3)=A(4) THEN LET W=500
280 LET S=S-10
290 IF W>0 THEN GOTO 350
300 PRINT AT 15,10;" PROFIT=";S;"P■"
310 REM
320 IF INKEY$="" THEN GOTO 320
330 PRINT AT 4,12;"■■■■■■■■■"
340 GOTO 130
350 PRINT AT 4,12;" WIN=";W;" P"
355 BEEP 0.5,38
360 LET S=S+W
370 LET W=0
380 GOTO 300
390 REM
400 FOR H=1 TO 4
410 PRINT AT 2,11;"■■■■■■■■■■■"
420 GOSUB 670
425 INK 2
430 PRINT AT 2,11;"NUDGE NOW "
435 INK 0
440 GOSUB 670
450 NEXT H
460 LET E=INT(RND*5)+2
470 FOR R=1 TO E
480 LET B=0
490 PRINT AT 3,12;E-R+1;" NUDGES"
500 LET G$=INKEY$
510 IF G$="" THEN GOTO 500
520 IF G$="1" THEN LET B=1
530 IF G$="2" THEN LET B=2
540 IF G$="3" THEN LET B=3
550 IF G$="4" THEN LET B=4
560 IF B=0 THEN GOTO 620
570 LET Z=A(B)
580 LET Y=INT(RND*6)+65
590 IF Y=Z THEN GOTO 580
600 LET A(B)=Y
605 INK Y-64
610 PRINT AT 9,6*B;CHR$(A(B))
615 INK 0
620 NEXT R
630 REM
640 PRINT AT 2,11;"■■■■■■■■■■■"
```

```
650 PRINT AT 3,12;"▬▬▬▬▬▬"
660 GOTO 250
670 FOR R=1 TO 20
680 NEXT R
690 RETURN
```

Full Program: RML

```
  5 GRAPH
 10 RANDOMIZE:PUT 12
 20 FOR T=1 TO 9
 30 FOR R=1 TO 31:PLOT R*2,(21-T)*3,204:NEXT R
 40 NEXT T
 50 REM
 60 REM
 70 REM
 80 FOR T=13 TO 21
 90 FOR R=1 TO 31:PLOT R*2,(21-T)*3,204:NEXT R
100 NEXT T
110 DIM A(7):A$=CHR$(204)+CHR$(204)+CHR$(204)+CHR$(204
)+CHR$(204):B$=A$+A$
120 LET S=0
130 REM
140 FOR R=1 TO 4
150 LET C=INT(RND(1)*6)+65
160 LET A(R)=C
170 NEXT R
180 LET W=0
190 REM
200 FOR R=1 TO 4
210 PLOT 12*R,30,A(R)
220 NEXT R
230 LET T=INT(RND(1)*6)+1
240 IF T=1 THEN GOTO 390
250 IF A(1)=A(2) THEN LET W=20
260 IF A(1)=A(2) AND A(2)=A(3) THEN LET W=100
270 IF W=100 AND A(3)=A(4) THEN LET W=500
280 LET S=S-10:PUT 12
290 IF W>0 THEN GOTO 350
300 PRINT TAB(10);"PROFIT=";S;"P "
310 REM
320 G=GET()
330 REM
340 GOTO 130
350 PRINT TAB(11);"WIN=";W;"P "
360 LET S=S+W
370 LET W=0
380 GOTO 300
390 REM
```

```
400 FOR H=1 TO 4
410 PLOT 22,57,B$
420 GOSUB 670
430 PLOT 22,57,"NUDGE NOW"
440 GOSUB 670
450 NEXT H
460 LET E=INT(RND(1)*5)+3
470 FOR R=1 TO E
480 LET B=0
490 PUT 12:PRINT TAB(8);"YOU HAVE ";E-R+1;" NUDGES"
500 G$=GET$()
510 REM
520 IF G$="1" THEN LET B=1
530 IF G$="2" THEN LET B=2
540 IF G$="3" THEN LET B=3
550 IF G$="4" THEN LET B=4
560 IF B=0 THEN GOTO 620
570 LET Z=A(B)
580 LET Y=INT(RND(1)*6)+65
590 IF Y=Z THEN GOTO 580
600 LET A(B)=Y
610 PLOT 12*B,30,A(B)
620 NEXT R
630 REM
640 PLOT 22,57,B$
650 REM
660 GOTO 250
670 FORR=1 TO 200
680 NEXT R
690 RETURN
```

Full Program: ZX81

```
 10 RAND
 20 FOR T=1 TO 8
 30 PRINT"                                "
 40 NEXT T
 50 PRINT
 60 PRINT
 70 PRINT
 80 FOR T=1 TO 10
 90 PRINT"                                "
100 NEXT T
110 DIM A(7)
120 LET S=0
130 FAST
140 FOR R=1 TO 4
150 LET C=INT(RND*6)+38
160 LET A(R)=C
170 NEXT R
```

```
180 LET W=0
190 PRINT
200 FOR R=1 TO 4
210 PRINT AT 9,6*R;CHR$(A(R))
220 NEXT R
230 LET T=INT(RND*6)
240 IF T=1 THEN GOTO 390
250 IF A(1)=A(2) THEN LET W=20
260 IF A(1)=A(2) AND A(2)=A(3) THEN LET W=100
270 IF W=100 AND A(3)=A(4) THEN LET W=500
280 LET S=S-10
290 IF W>0 THEN GOTO 350
300 PRINT AT 15,10;" PROFIT=";S;"P"
310 SLOW
320 IF INKEY$="" THEN GOTO 320
330 PRINT AT 4,12;"          "
340 GOTO 130
350 PRINT AT 4,12;" WIN=";W;" P"
360 LET S=S+W
370 LET W=0
380 GOTO 300
390 SLOW
400 FOR H=1 TO 4
410 PRINT AT 2,11;"          "
420 GOSUB 670
430 PRINT AT 2,11;"NUDGE NOW "
440 GOSUB 670
450 NEXT H
460 LET E=INT(RND*5)+2
470 FOR R=1 TO E
480 LET B=0
490 PRINT AT 3,12;E-R+1;" NUDGES"
500 LET G$=INKEY$
510 IF G$="" THEN GOTO 500
520 IF G$="1" THEN LET B=1
530 IF G$="2" THEN LET B=2
540 IF G$="3" THEN LET B=3
550 IF G$="4" THEN LET B=4
560 IF B=0 THEN GOTO 620
570 LET Z=A(B)
580 LET Y=INT(RND*6)+38
590 IF Y=Z THEN GOTO 580
600 LET A(B)=Y
610 PRINT AT 9,6*B;CHR$(A(B))
620 NEXT R
630 FAST
640 PRINT AT 2,11;"          "
650 PRINT AT 3,12;"          "
660 GOTO 250
670 FOR R=1 TO 4
680 NEXT R
690 RETURN
```

Writing Instructions

Giving your games instructions for the player will add a very professional touch and also save you having to explain it all to any of your friends who want to have a go.

First of all, let's give your game a title and then ask the player whether he or she would like to have instructions. This should all happen before the main program loop and there will probably be a logical place to slot it in on each program. These line numbers are intended as a guide only.

BBC

```
   10 PRINT TAB(15,11);"TITLE"
   20 PRINT TAB(1,14);"DO YOU WANT TO KNOW HOW TO PLAY?
(Y/N)"
   30 INPUT P$
   40 IF P$="Y" THEN GOSUB 1000
```

RML

```
   10 PLOT 30,30,"TITLE"
   20 PLOT 2,21,"DO YOU WANT TO KNOW HOW TO PLAY? (Y/N)"
   30 INPUTP$
   40 IF P$="Y" THEN GOSUB 1000
```

Spectrum and ZX81

```
   10 PRINT AT 11,15;"TITLE"
   20 PRINT AT 14,1;"DO YOU WANT TO KNOW HOW TO PLAY? (Y
/N)"
   30 INPUT P$
   40 IF P$="Y" THEN GOSUB 1000
```

Then at line 1000 you can start writing your instructions. You should give an exciting description of the game, tell the player exactly what the aim of the game is, list the keys the

player has to use, and add anything else you think might be useful. Spend a little time making your instructions look good and well-spaced. Try to centre them on the screen and also watch out for awkward breaks in words at the ends of lines. You can cure this fault easily, just by adding spaces at the right places in your listing – it makes your listing look a bit odd but looks great on the screen! If you have colour and sound, make full use of them to get a really spectacular look.

When you have finished your instructions, you have to remember to give the player time to read them. To do this, you can either introduce a delay loop, like this:

```
1500 FOR R=1 TO 1000
1510 NEXT R
```

which will pause for 1,000 loops (the length of time will vary enormously from micro to micro – just experiment a bit), or you can get the computer to wait for the player to press a key before going any further:

BBC

```
1500 G=GET
```

RML, Spectrum and ZX81

```
1500 IF INKEY$="" THEN GOTO 1500
```

Finally, you must remember to make the last line in your instruction subroutine RETURN, which will send the computer back to its right place in the main program.

Writing Your Own Programs

As you will have seen from the games in this book, it is possible to work out some really quite exciting or puzzling games from very simple elements. The important thing to remember when starting to write your own programs is to keep your ideas as simple as possible. As you become more used to the way programming works, you will be able to build on your earlier ideas and become more ambitious.

Almost the biggest problem you face is thinking of an idea for a game. Decide first what sort of game you want to try. Will it be a shooting game, a bat-and-ball game, a chasing game or a puzzle? Once you have decided, work out roughly on paper the main elements of the game in order; then work on them one at a time, in chunks, as we have done in this book. You might find writing a flow chart useful. Sometimes you will be able to test little bits by giving some of your variables defined values. This won't run the game but might give you a visual check.

Try to keep your main program loops as simple as possible and make the computer jump out to a subroutine for anything at all complicated, using IF . . . THEN GOSUB . . . . Remember that subroutines must end with RETURN, otherwise the computer will arrive there and won't be able to get back to the main loop! Give your subroutines high numbers, starting with 1000, 2000, 3000, etc. This makes them easier to identify and also keeps them well out of the way of the main program, leaving you plenty of room to expand into. There's nothing worse than having to renumber a whole listing simply because you have run out of lines to use! It is also a good idea to use REM to label your subroutines. If the first line of each of your subroutines begins with 'REM move bat routine', 'REM check for hit routine' and so on, you will find it much easier to remind yourself which bit of your program does what. The computer will always ignore a line beginning with REM.

What do you do when it doesn't work?

Unless you are a naturally gifted programmer, or extremely lucky, it is almost certain that your new program won't work the first few times. In fact, programming wouldn't be nearly as much fun if you did get it right each time. Tracking down the bugs in a program can be just as demanding and satisfying as playing the game at the end.

Usually your computer will give you some idea of what is wrong through an error code. You will have to check your manual for the exact meanings of these.

Don't forget that the computer cannot do anything you haven't told it to do. It won't be able to make the leaps of imagination your brain can make, because it only thinks step by step and you have to show it which steps to take.

Finally, be patient with your computer. If anything goes wrong, or your program won't work, it is not the computer's fault – it's yours. Sometimes the best thing to do is to save what you have done so far and go away. A few hours' rest from the program may give you some new ideas for making it work.

Program Notes

Fighting Fantasy Gamebooks

1. THE WARLOCK OF FIRETOP MOUNTAIN

Steve Jackson and Ian Livingstone

With this book you will need two dice, a pencil and an eraser: then, armed with these tools, you can become the hero in a perilous quest to find the Warlock's hidden treasure. It is hidden deep within a dungeon which *you* must explore, populated with a multitude of underworld monsters which *you* must fight and kill – or be killed in the attempt.

2. THE CITADEL OF CHAOS

Steven Jackson

The Citadel of Chaos holds a dark and dangerous peril for anyone foolhardy enough to venture through its gruesome gates. And yet venture you must, for your mission lies at the heart of the Citadel, with the dread sorcerer, Balthus Dire.

3. THE FOREST OF DOOM

Ian Livingstone

Only the brave would risk an encounter with the unknown perils that lurk in the murky depths of Darkwood Forest. Yet there is no alternative, for your quest is a desperate race against time to find the missing pieces of the legendary Hammer of Stonebridge – fashioned by Dwarfs to protect the villagers of Stonebridge against their ancient doom.

4. STARSHIP TRAVELLER

Steve Jackson

Sucked through the appalling nightmare of the Seltsian Void, the starship *Traveller* emerges at the other side of the black hole into an unknown universe. YOU are the captain of the *Traveller* and her fate lies in your hands. Will you be able to discover the way back to Earth from the alien peoples and planets you encounter, or will you and your crew be doomed to roam uncharted space forever?

5. CITY OF THIEVES

Ian Livingstone

Terror stalks the night as Zanbar Bone and his bloodthirsty Moon Dogs hold the prosperous town of Silverton to ransom. YOU are an adventurer, and the merchants of Silverton turn to you in their hour of need. Your mission takes you along dark, twisting streets where thieves, vagabonds and creatures of the night lie in wait to trap the unwary traveller. And beyond lies the most fearsome adventure of them all – the tower stronghold of the infamous Zanbar Bone!

Heard about the Puffin Club?

. . . it's a way of finding out more about Puffin books and authors, of winning prizes (in competitions), sharing jokes, a secret code, and perhaps seeing your name in print! When you join you get a copy of our magazine, *Puffin Post*, sent to you four times a year, a badge and a membership book.
For details of subscription and an application form, send a stamped addressed envelope to:

The Puffin Club Dept A
Penguin Books Limited
Bath Road
Harmondsworth
Middlesex UB7 0DA

and if you live in Australia, please write to:

The Australian Puffin Club
Penguin Books Australia Limited
P.O. Box 257
Ringwood
Victoria 3134

- ***games for BBC (models A and B), RML (380Z and 480Z), ZX Spectrum and ZX81 micros***
- ***each program explained in easy-to-follow steps, with suggestions for variations you can make***
- ***a guide to help you go on to devise your own programs***

In 1981 Patrick Bossert showed the world how to crack the cube in his bestselling book *You Can Do the Cube.* Now he has turned his attention to the microcomputer and devised a great collection of games listings.

There is a death-defying bomb run, a hair-raising car race, a bat-and-ball game, fiendish puzzles to work out, and many more.

All the games require lightning reactions, quick thinking or good powers of observation. All will give hours of fun – and all will help you understand your computer that little bit better!

Cover illustration by John Farman

Photograph of Patrick Bossert by Geoff Howes
Photograph of Philippa Dickinson by Jeremy Mulcaire

U.K. £1.50
AUST. $4.50
(recommended)
CAN. $2.95

A PUFFIN BOOK

ISBN 0 14
03.1667 1